

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A188 094

DTIC FILE COPY

14

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 1536900		5. MONITORING ORGANIZATION REPORT NUMBER(S) <i>Distribution is unlimited</i>	
6a. NAME OF PERFORMING ORGANIZATION University of Colorado	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION ONR	
6c. ADDRESS (City, State and ZIP Code) University of Colorado, Box 425 Boulder CO 80309		7b. ADDRESS (City, State and ZIP Code) University of New Mexico Albuquerque, NM 87131	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION ONR	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-K-0425	
8c. ADDRESS (City, State and ZIP Code) Arlington, VA 22217		10. SOURCE OF FUNDING NOS.	
		PROGRAM ELEMENT NO.	PROJECT NO.
11. TITLE (Include Security Classification) Time Varying Controller Design for Robot Manipulator Control		NOV 16 1987	
12. PERSONAL AUTHOR(S) M.Y. Wu/N.M. Arbouz/M.F. Chonika			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 7/1/84 TO 6/30/87	14. DATE OF REPORT (Yr., Mo., Day) 87, 8, 25	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP SUB. GR.		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This report studies the design of time-varying controller for robot manipulator control. Piece-wise constant time-varying controller is shown to be capable of producing desired fast response without causing overshoot. Such response can not be achieved with time-invariant controller. Nonlinear feedback linearization method is used to transform nonlinear robot equation into decoupled sets of second-order linear equations so that time-varying controller can be designed. Stability of time-varying system and time-varying control based on learning algorithm are also presented</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL	

Abstract/Preface

This report presents some results for studies done under the ONR contract N00014-84-K-0425. The work was aimed at developing time-varying controller design theory for potential applications to robot manipulator control. The reports comprise of the following four parts:

Part 1) Robot Manipulator Controller Design: It was demonstrated that time-varying controllers can produce more desirable performances than those of time-invariant ones.

Part 2) Stability of Time-Varying System: An important observation was made regarding stability in the design of time-varying systems. Implications of magnitude of change, and rate of change of parameter variations on stability are noted.

Part 3) Learning Controller Design: One particular time-varying controller design approach that is of most interest to robot control is the learning control method. This part presents some preliminary results on learning control theory.

Part 4) A collection of papers on work which are in part supported by this ONR contract.

✓
Added in file

A-1



Part 1

Robot Manipulator Controller Design

CHAPTER I

INTRODUCTION

1.1 Introduction

Industrial robots which are defined as computer controlled mechanical manipulators have become increasingly important in industrial automation in recent years. They can be programmed to perform the tasks, without human intervention, of arc welding, paint spraying, assembly, foundry operation, etc.

A manipulator can be described as a series of links connected at joints. Typically, they have three to six joints (three to six degrees of freedom) with a gripper or end effector. Each joint is driven by an actuator which is connected to sensing devices.

The control system design for robot manipulators is basically the problem of controlling a multi-input nonlinear system. The general objective is to achieve a very accurate, fast and smooth tracking while rejecting a broad class of disturbances, including parameter variations.

The simplest form of control used for manipulators is the open loop control ([1]-[3]), where the entire input sequence is predetermined and is applied regardless of any errors which develop. The trajectory is preplanned or prerecorded and the

inputs do not depend on output measurements. Disturbance rejection and accurate path tracking can only be achieved by making the robot extremely rigid. This approach which is commonly used today implies precise gear trains and actuators as well as very strong structural members. The speed of such systems is generally limited by the force-producing capacity and speed of the actuators, and by the excitation of high-frequency structural modes of the linkage.

Linear feedback controls have also been designed for manipulators. The most widely used method is the independent joint control ([4],[5]), where basically each joint is independently controlled by a linear PD controller. Gravity compensation is also provided. The 'computed torque' technique is an illustration of the independent joint control. We will present it in detail in chapter IV.

A pseudo-linear feedback law with nonlinear pre- and post-processing of measurements was developed by Raibert and Craig [6]. In practice, this kind of linear control is easy to implement, works relatively well and is reliable. However, adequate disturbance accommodation requires high-power actuators and this technique becomes less effective when high speed and accurate tracking are required.

In order to improve performances (speed, accuracy, transient response, etc.), more sophisticated methods have been developed in recent years.

Adaptive control schemes based on the model reference principle have been proposed ([7]-[9]). Lee and Chung [10]

presented an adaptive control based on the first order linearization of the nonlinear dynamic equations. In general, these adaptive strategies are characterized by complex algorithms which may lead to complexity comparable to that required for real time computation of inverse kinematics.

Nonlinear feedback laws have also been proposed by many authors ([11],[12]). Young [13] used sliding mode theory to develop nonlinear switching feedbacks.

Since the mid-1970's, the technique of linearizing a nonlinear system via nonlinear feedbacks has been used and developed into what is known today as 'differential geometric control theory' ([14]-[25]). The earlier work (1976) was conducted by Hemami and Camano [21] who applied this technique to a simple locomotion system which resulted in uncoupled subsystems. In 1982, Freund [22] through a 'global nonlinear feedback law' obtained uncoupled second order systems. This method is based on partitioning the robot dynamic equations. But one drawback of this method is that the number of inputs should equal the number of outputs.

Recently, a theorem giving necessary and sufficient conditions to linearize a nonlinear system via a coordinate transformation and nonlinear feedbacks has been given. This result is due in part to Brockett [23], Jakubczyk and Respondek [24] and finalized by Hunt, Su and Meyer ([18],[25]).

In this thesis our main work will be, using this feedback linearization theorem, to show that a given n -joint robot

manipulator can be linearized through a coordinate transform and nonlinear feedbacks. The resulting linear system is composed of n uncoupled second order time invariant subsystems. The control problem is then reduced to controlling these n uncoupled second order subsystems. A general expression of the explicit feedback controls will be derived. Two examples will be given, a two and three joint robot manipulator, with accompanying computer simulation study.

1.2 Organization of the Thesis

In chapter II, we discuss the robotics problem in terms of trajectory planning and control approach to be followed for a given robot task.

Chapter III deals with the manipulator dynamics. To design a robot control system, one needs a mathematical model of the robot. Such a model can be obtained by Newton-Euler or Lagrangian mechanics.

Chapter IV is the main part of the thesis. We present first a robot controller using the 'computed torque' technique. Then, after giving some theoretical background needed for further developments, we state the feedback linearization theorem [25]. Using this theorem, we linearize the manipulator dynamic equations and derive a general form of the nonlinear feedback law. Then, we illustrate by two examples with computer simulations.

Finally, the summary and conclusions are given in chapter V.

CHAPTER II

TRAJECTORY PLANNING AND CONTROL

The basic problem in robotics is planning trajectories to solve for some specified task and then controlling the robot to achieve those trajectories.

The trajectory planning consists of computing a desired sequence of positions, velocities and accelerations of some point which is usually the robot hand. This is in fact the so called kinematics problem which will be discussed in the next section.

The control strategy to be adopted depends on the nature of the specified task itself. For example, if the robot is permitted to travel between the initial and final positions, a simple point to point control is adequate. In this chapter we will give a brief survey of these control strategies.

2.1 Kinematics

Kinematics of a manipulator ([26],[27]) can be defined as being the position, velocity and acceleration relationships among the links of the manipulator.

In planning a trajectory, one is primarily interested in the position of the hand with respect to the work space, which is called the hand space of the manipulator. A hand configuration in hand space consists of position described by a vector \vec{P} and

orientation described by three orthogonal vectors: the approach vector \vec{a} , the orientation vector \vec{o} and the normal vector \vec{n} .

A robot task is naturally specified in terms of its hand configuration in hand space. It is a transformation (matrix: $(\vec{n}(t), \vec{o}(t), \vec{a}(t), \vec{p}(t))$), called the forward kinematics transform which relates the hand frame to the robot base frame.

To achieve the desired configuration, one has to command the joint actuators. To do so, we must be able to find the corresponding joint coordinates (in joint space) from the desired hand configuration. This inverse problem is referred to as the inverse kinematics transform, or arm solution.

The direct kinematics has a straightforward solution, whereas problems can occur when computing the inverse kinematics. The solution may not be unique and singularities may occur, depending on the geometrical configuration of the arm.

2.2 Control

As mentioned previously, the control strategy to be considered depends on the assigned robot task. These strategies are classified as follows:

2.2.1 Point to point control

If there are no path constraints, if the work space is free and if coordination with external moving objects (e.g. conveyors) is not required, positional control can be used to ensure that the hand passes through the specified corner points of the path. No control over position is required between points. The path of the

hand in such control schemes is unpredictable and the robot exhibits a tendency to stop at each point.

In many cases we require that the hand moves smoothly along a prescribed path (path tracking). This involves many computations of corresponding desired joint coordinates (through the inverse kinematic transform). Two cases occur here: off-line path control, in which computations are performed before the motion starts and on-line path control, in which calculations are performed in real time.

2.2.2 Off-line path control

If the work space is free and no external coordination is needed, the hand path and the corresponding desired joint coordinates can be specified before the motion is to start. To accomplish a smooth motion (smooth accelerations), some techniques are available, for example the path control polynomials technique [26].

2.2.3 On-line path control

When external coordination is needed, then path points and desired joint coordinates have to be computed on-line. This constraint is computationally very difficult, and such a strategy is in practice used when accurate path tracking is important and the manipulator moves slowly. There are control techniques which compromise between full on-line path control and point to point control, in order to achieve smooth and accurate tracking (e.g., joint interpolated control ([26],[28])).

2.2.4 Collision free path control

If the work space is changing and not free, then collision free paths must be followed. This makes path control very complex if on-line path control is required.

2.2.5 Force control

In some situations, the manipulator is constrained by external positional constraints. Two common situations can happen: guarded motion, when the manipulator is about to contact a surface; and compliant motion, when the manipulator is in continuing contact with a surface. In those cases, one has to control the forces instead of the positions and we will be relating forces in hand space to torques (and/or forces) in joint space.

Some controllers can simultaneously control forces along certain coordinate axes and control positions along the remaining axes. They are referred to as hybrid controllers.

CHAPTER III

DYNAMICS OF A ROBOT MANIPULATOR

A robot manipulator is a mechanical structure which consists of a series of links connected at joints. When several joints move simultaneously, the motion and the torque applied at one joint have a dynamic effect on the motion at other joints. This results in high coupling among the joints and makes the overall system dynamics very complex.

For purposes of dynamic control, one needs a mathematical model of the manipulator. Such a model can be obtained by deriving the robot equations of motion using either the Lagrangian or the Newton-Euler approach [29]. The resulting dynamic equations for a n -joint manipulator are highly nonlinear and coupled. They have the following form:

$$D(q)\ddot{q} + H(q, \dot{q}) + G(q) = \tau \quad (3.1)$$

where q is an $(n \times 1)$ vector of the actual joint positions (n joints).

$D(q)$ is an $(n \times n)$ inertial forces matrix.

$H(q, \dot{q})$ is an $(n \times 1)$ coriolis and centrifugal force vector.

$G(q)$ is an $(n \times 1)$ gravitational force vector.

τ is a $(n \times 1)$ generalized input vector (torque and/or force).

For a six-joint manipulator, computing these equations is a very

difficult task. The result is hundreds of algebraic terms which makes the on-line computation of the control torques a major problem in robot control.

Next, we derive the dynamic equations of a two joint and a three joint manipulator that will be used later for purposes of control.

3.1 Dynamic Equations of a 2-Joint Manipulator

We first consider the two-rotational-joint robot manipulator of Figure 3.1, where the first link has a length l_1 and mass m_1 , the second link has a length l_2 and mass m_2 . Both masses are considered to be centered at the link mid-points. The load of mass m_L is placed at the end of the second link. Using polar coordinates the potential energy can be expressed as:

$$V = m_1 g \frac{l_1}{2} \sin \theta_1 + m_2 g (l_1 \sin \theta_1 + \frac{l_2}{2} \sin(\theta_1 + \theta_2)) + m_L g (l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)) \quad (3.2)$$

The kinetic energy, which is the sum of the kinetic energies of link 1 (m_1), link 2 (m_2) and the load M_L , is:

$$\begin{aligned} K = & \frac{1}{8} m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{8} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ & + \frac{1}{2} m_2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 + \frac{1}{2} m_L l_1^2 \dot{\theta}_1^2 \\ & + \frac{1}{2} m_L l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_L l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 \quad (3.3) \end{aligned}$$

where θ_1, θ_2 are the joint angles and $\dot{\theta}_1, \dot{\theta}_2$ are the corresponding angular velocities.

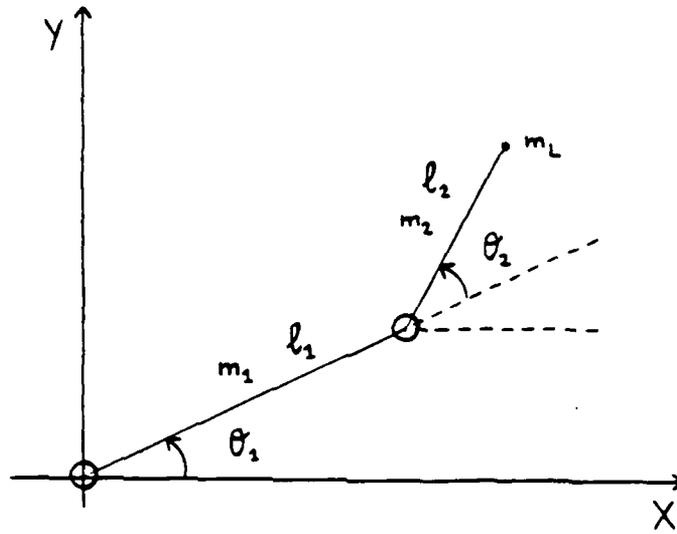


Figure 3.1. Two-joint manipulator

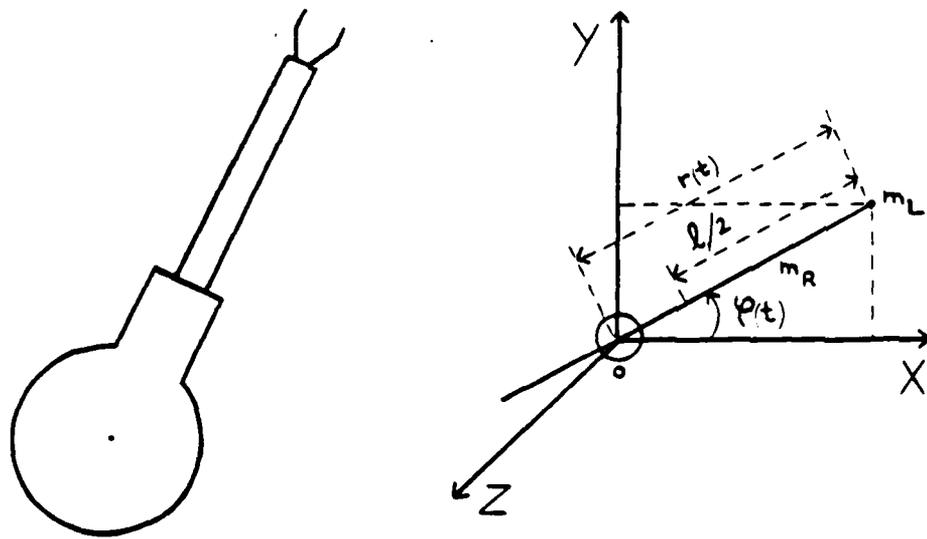


Figure 3.2. Three-joint manipulator

The Lagrangian $L = K - V$ is then,

$$\begin{aligned}
 L = & \left(\frac{m_1}{8} + \frac{m_2}{2} + \frac{m_L}{2}\right)l_1^2\dot{\theta}_1^2 + \left(\frac{m_2}{8} + \frac{m_L}{2}\right)l_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\
 & + \left(\frac{m_2}{2} + m_L\right)l_1l_2(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2)\cos\theta_2 - \left(\frac{m_1}{2} + m_2 + m_L\right)gl_1\sin\theta_1 \\
 & - \left(\frac{m_2}{2} + m_L\right)gl_2\sin(\theta_1 + \theta_2)
 \end{aligned} \tag{3.4}$$

The dynamic equations are obtained from the Lagrangian equations:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = T_i \quad i = 1, 2 \tag{3.5}$$

where T_i is the torque (or force) applied at each joint. The final dynamic equations are:

$$\begin{aligned}
 T_1 = & \left[\left(\frac{m_1}{4} + m_2 + m_L\right)l_1^2 + \left(\frac{m_2}{4} + m_L\right)l_2^2 + 2\left(\frac{m_2}{2} + m_L\right)l_1l_2\cos\theta_2 \right] \ddot{\theta}_1 \\
 & + \left[\left(\frac{m_2}{4} + m_L\right)l_2^2 + \left(\frac{m_2}{2} + m_L\right)l_1l_2\cos\theta_2 \right] \ddot{\theta}_2 \\
 & - \left(\frac{m_2}{2} + m_L\right)l_1l_2\sin\theta_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 + \left(\frac{m_1}{2} + m_2 + m_L\right)gl_1\cos\theta_1 \\
 & + \left(\frac{m_2}{2} + m_L\right)gl_2\cos(\theta_1 + \theta_2) .
 \end{aligned} \tag{3.6}$$

$$\begin{aligned}
 T_2 = & \left[\left(\frac{m_2}{4} + m_L\right)l_2^2 + \left(\frac{m_2}{2} + m_L\right)l_1l_2\cos\theta_2 \right] \ddot{\theta}_1 \\
 & + \left(\frac{m_2}{4} + m_L\right)l_2^2\ddot{\theta}_2 + \left(\frac{m_2}{2} + m_L\right)l_1l_2\dot{\theta}_1^2\sin\theta_2 \\
 & + \left(\frac{m_2}{2} + m_L\right)gl_2\cos(\theta_1 + \theta_2) .
 \end{aligned} \tag{3.7}$$

3.2 Dynamic Equations of a 3-Joint Manipulator

Consider now the three joint manipulator of Figure 3.2. It consists of one rotational joint which rotates in the (x-y) plane (joint variable ϕ). It also has two prismatic joints. One allows the hand to extend in the (x-y) plane (joint variable r) while the other lets the hand translate along the z-axis (joint variable z). The arm has a length l and a mass m_R . The load with a mass m_L is placed at the end of the arm as shown in Figure 3.2. We suppose that $r \geq l/2$.

In the cylindrical coordinates the potential energy is:

$$V = m_L g r \sin \phi + (r - \frac{l}{2}) m_R g \sin \phi \quad (3.8)$$

The kinetic energy is:

$$\begin{aligned} K = & \frac{1}{2} m_L (\dot{r}^2 + (r\dot{\phi})^2 + \dot{z}^2) + \frac{1}{2} m_R \left(\frac{r}{l}\right) \left[\left(\frac{\dot{r}}{2}\right)^2 + \left(\frac{r\dot{\phi}}{2}\right)^2 + \dot{z}^2 \right] \\ & + \frac{1}{2} m_R \left(\frac{l-r}{l}\right) \left[\left(\frac{\dot{r}}{2}\right)^2 + \left(\frac{l-r}{2} \dot{\phi}\right)^2 + \dot{z}^2 \right] \end{aligned} \quad (3.9)$$

The lagrangian L is then,

$$\begin{aligned} L = & \frac{1}{2} m_L (\dot{r}^2 + r^2 \dot{\phi}^2 + \dot{z}^2) + \frac{1}{2} m_R \left(\frac{r}{l}\right) \left[\left(\frac{\dot{r}}{2}\right)^2 + \left(\frac{r\dot{\phi}}{2}\right)^2 + \dot{z}^2 \right] \\ & + \frac{1}{2} m_R \left(\frac{l-r}{l}\right) \left[\left(\frac{\dot{r}}{2}\right)^2 + \left(\frac{l-r}{2} \dot{\phi}\right)^2 + \dot{z}^2 \right] - m_L g r \sin \phi \\ & - (r - \frac{l}{2}) m_R g \sin \phi . \end{aligned} \quad (3.10)$$

If we denote by F_R the force applied by the actuator along the r direction, by F_z the force applied by the second

prismatic joint along the z-axis, and by T the torque applied by the rotational joint; we obtain the following dynamic equations:

$$F_R = (m_L + \frac{1}{4} m_R) \ddot{r} - (m_L + \frac{3}{4} m_R) r \dot{\phi}^2 + \frac{3}{8} m_R \ell \dot{\phi}^2 + (m_L + m_R) g \sin \phi . \quad (3.11)$$

$$T = 2(m_L + \frac{3}{4} m_R) r \dot{r} \dot{\phi} + (m_L + \frac{3}{4} m_R) r^2 \ddot{\phi} - \frac{3}{4} m_R \ell \dot{r} \dot{\phi} + \frac{1}{4} m_R \ell (\ell - 3r) \ddot{\phi} + (m_L + m_R) g r \cos \phi - \frac{\ell}{2} m_R g \cos \phi \quad (3.12)$$

$$F_z = (m_L + m_R) \ddot{z} \quad (3.13)$$

CHAPTER IV

ROBOT CONTROLLER DESIGN

In this chapter we present two different control concepts. In section 4.1 we present an independent joint control method, the 'computed torque' technique.

In section 4.2 we propose a nonlinear control approach based on feedback linearization. A control algorithm with explicit nonlinear feedback is derived. We also illustrate by two design examples with computer simulations to evaluate the performance of the proposed control method.

4.1 Controller Design with the 'Computed Torque' Technique

Most of the control approaches found in today's commercial robots use the method of independent joint control ([4],[5]). An illustration of these methods is the 'computed torque' technique ([30],[31]), also called the 'inverse problem' technique ([32],[33]). It is basically a linear proportional and derivative control law.

As seen in chapter III, the actual equations of motion of a robot are in the form:

$$D(q)\ddot{q} + H(q,\dot{q}) + G(q) = \tau \quad (4.1)$$

The principle of the 'computed torque' technique is as follows. Let $D_c(q)$, $H_c(q, \dot{q})$ and $G_c(q)$ be the computed counterparts of the actual $D(q)$, $H(q, \dot{q})$ and $G(q)$. Let the control τ (force or torque) be:

$$\tau = D_c(q) [\ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})] + H_c(q, \dot{q}) + G_c(q) \quad (4.2)$$

where q_d , \dot{q}_d and \ddot{q}_d are respectively the desired joint position, velocity and acceleration vectors.

K_p and K_v are constant scalar feedback gains (PD action). If we assume that the computed D_c , H_c and G_c are equal to their actual counterparts, i.e.:

$$\begin{aligned} D_c(q) &= D(q) \\ H_c(q, \dot{q}) &= H(q, \dot{q}) \\ G_c(q) &= G(q) \end{aligned} \quad (4.3)$$

then from (4.1) and (4.2) we obtain:

$$(\ddot{q}_d - \ddot{q}) + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q}) = 0 \quad (4.4)$$

If we let $e_q = q_d - q$ be the joint position error, then (4.4) becomes:

$$\ddot{e}_q + K_v \dot{e}_q + K_p e_q = 0 \quad (4.5)$$

The control problem is then reduced to assigning poles with negative real parts for (4.5) such that the error e_q approaches zero asymptotically. One should note that the convergence relies on the validity of (4.3).

From (4.5), the component e_{q_i} of the error vector e_q has the following characteristic equation:

$$s^2 + K_v s + K_p = s^2 + 2 \zeta \omega_n s + \omega_n^2 = 0 \quad (4.6)$$

where

$\omega_n = \sqrt{K_p}$ is the natural frequency.

$\zeta = \frac{K_v}{2\sqrt{K_p}}$ is the damping ratio.

In almost any robot application overshoot is to be avoided. In this case, the fastest response with no overshoot is the critically damped one corresponding to ζ equals to one. Hence,

$$K_v = 2\sqrt{K_p}$$

and the error time response is:

$$e_{q_i}(t) = c_1 e^{-\sqrt{K_p} t} + c_2 t e^{-\sqrt{K_p} t} \quad (4.7)$$

where c_1, c_2 are constants.

From (4.7) one can say that the larger the values of the feedback gains K_p and K_v , the faster the asymptotic convergence of $e_{q_i}(t)$.

As an example we will use this technique to control the three joint manipulator of figure 3.2.

Example 4.1.1. Three joint manipulator:

From the dynamic equations (3.11), (3.13) and the control law as given by (4.2), the actual control torque T and control forces F_R, F_Z are:

$$F_R = \left(m_L + \frac{m_R}{4}\right)(\ddot{r}_d + K_p(r_d - r) + K_v(\dot{r}_d - \dot{r})) - \left(m_L + \frac{3}{4}m_R\right)r\dot{\phi}^2 + \frac{3}{8}m_R l \dot{\phi}^2 + (m_L + m_R)g \sin \phi. \quad (4.8)$$

$$T = \left(\left(m_L + \frac{3}{4}m_R\right)r^2 + \frac{1}{4}m_R l(l-3r)\right)(\ddot{\phi}_d + K_p(\phi_d - \phi) + K_v(\dot{\phi}_d - \dot{\phi})) + 2\left(m_L + \frac{3}{4}m_R\right)r\dot{r}\dot{\phi} - \frac{3}{4}m_R l\dot{r}\dot{\phi} + (m_L + m_R)gr \cos \phi - m_R \frac{l}{2}g \cos \phi. \quad (4.9)$$

$$F_Z = (m_L + m_R)(\ddot{z}_d + K_p(z_d - z) + K_v(\dot{z}_d - \dot{z})) \quad (4.10)$$

where $q = (r, \phi, z)^T$ and \dot{q} are the actual joint position and velocity vectors, $q_d = (r_d, \phi_d, z_d)^T$ and \dot{q}_d are the desired joint position and velocity vectors, $\ddot{r}_d, \ddot{\phi}_d, \ddot{z}_d$ are the desired accelerations. K_p, K_v are constant feedback gains.

A computer simulation was then conducted. In the first part of the simulation, we simulated the motion of the end effector from one initial path point to the next desired path point, or equivalently through the inverse kinematics, from an initial joint state to a desired one.

Initial joint state ($r = 1$ m, $\phi = 0$ rd, $z = -1$ m), and desired one ($r = 0.5$ m, $\phi = -0.6$ rd, $z = -0.2$ m) were arbitrarily chosen. For the entire simulation the feedback gains K_p and K_v were chosen to maintain a critically damped response, i.e.

$$K_v = 2\sqrt{K_p}.$$

Figure 4.1, 4.2 and table 4.1 show the simulation results for $K_p = 5000$. A steady state error is present, due to the fact that (4.5) represents a type 0 system. High torques are also required for the motion.

Table 4.2 gives the steady state errors and the maximum value of force F_z (F_z showed higher values than T and F_R) as a function of the feedback gain K_p .

One notes that as K_p increases, the steady state errors and the convergence time decrease. However, the torque values at the beginning of the motion increase, which means more powerful actuators are needed. Indeed, the position error e_q and its derivative \dot{e}_q are maximal at the beginning of the motion. The inertial term is the dominant part in the expressions of the torques given by (4.2). This implies that high values of K_p and K_v may result in high torques until e_q and \dot{e}_q are small enough to cancel out the effect of such K_p and K_v . Hence, one should compromise between accuracy, speed and maximum allowable torques.

One way to maintain good accuracy, fast response and smaller torques is with time varying feedback gains. One can first put a bound on the maximum permissible torques and starts with small values of K_p and K_v . As the errors e_q and \dot{e}_q are

Table 4.1. Simulation of joint motions from (1,0,-1) to (0.5,-0.6,-0.2) with $K_p = 5000$.

THREE JOINT ROBOT MANIPULATOR-COMPUTED TORQUE TECHNIQUE
CONSTANT FEEDBACK GAINS.

TIME	R	PHI	Z
0	1.00000	0	-1.00000
.10000	.51751	- .58181	- .19139
.20000	.51418	- .58582	- .18582
.30000	.51418	- .58582	- .18582
.40000	.51418	- .58582	- .18582
.50000	.51418	- .58582	- .18582
.60000	.51418	- .58582	- .18582

Table 4.2. Steady state position error, maximum force F_z and convergence time, as a function of the gain K_p .

Feedback gain K_p	Δr (mm)	$\Delta \phi$ (mrd)	Δz (mm)	$(F_z)_{max}$ (N)	Convergence time (s)
400	51	51	51	4762	0.72
1000	32	32	32	11646	0.44
2000	22	22	22	23029	0.33
5000	14	14	14	56993	0.22
10000	10	10	10	113403	0.15

THREE JOINT MANIPULATOR

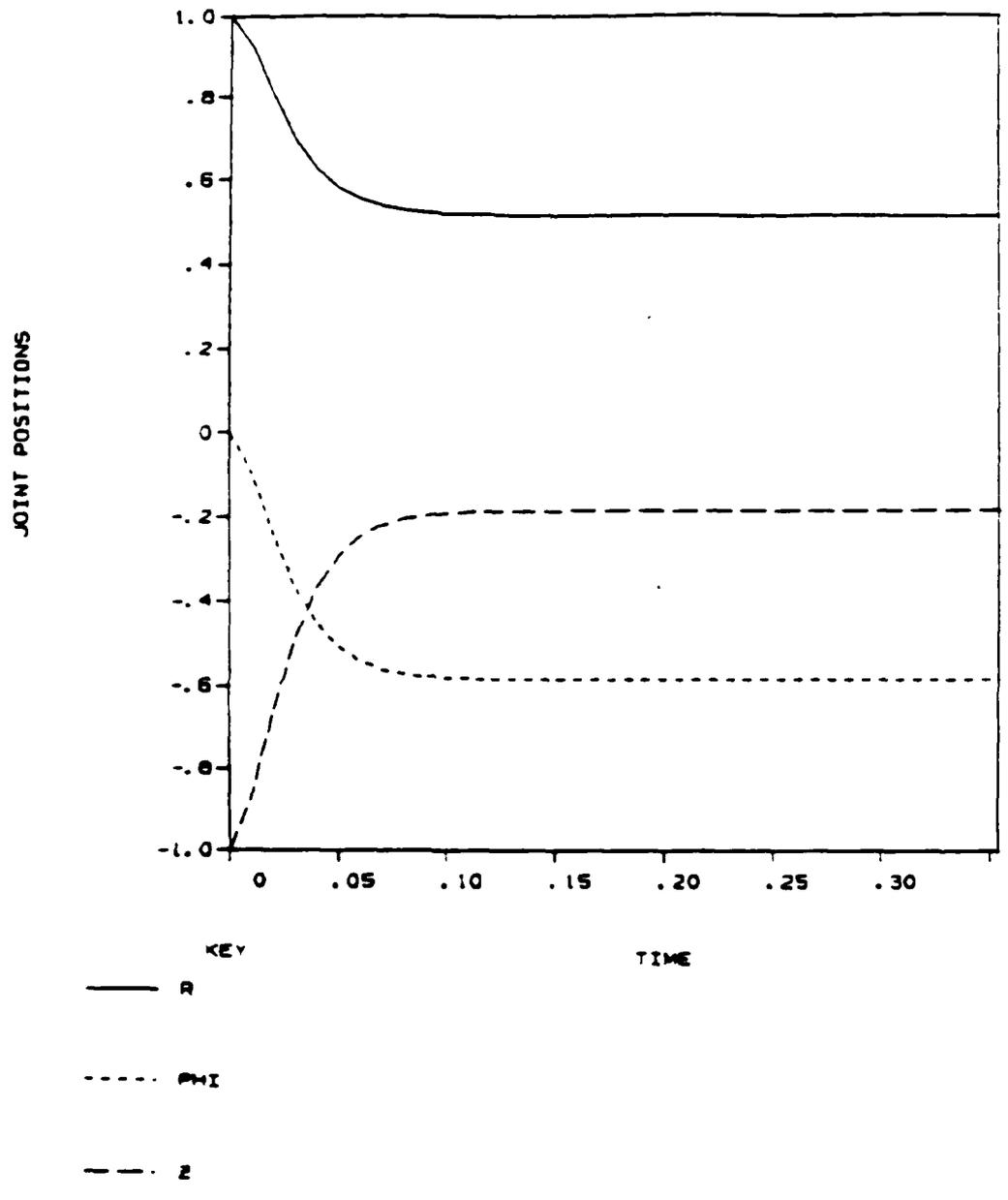


Figure 4.1. 'Computed torque' technique. Simulation of joint motions from (1,0,-1) to (0.5,-0.6,-0.2).

THREE JOINT MANIPULATOR

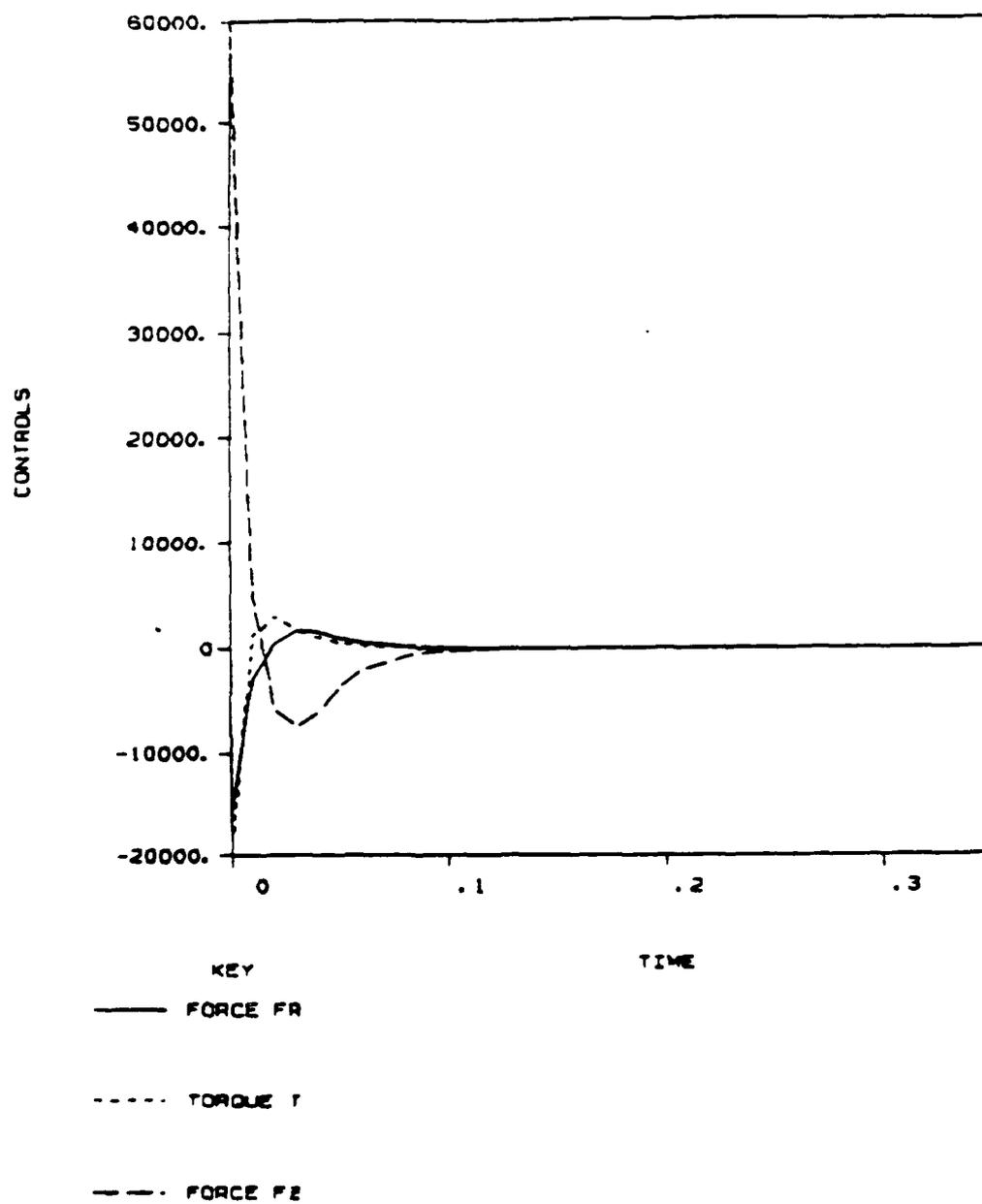


Figure 4.2. 'Computed torque technique'. Controls.

decreasing, K_p and K_v are increased as long as the torque limits are not exceeded.

Using this technique and increasing K_p , K_v in a piecewise fashion [34], the previous hand motion of the three joint manipulator was again simulated. The maximum allowable torque (force) was set to be 5000 Nm (N). Simulation results are given in table 4.3 and figures 4.3 - 4.5. The results show a steady state error of 4 mm for joint variables r , z and 4 mrd for ϕ . The convergence time of the motion is 0.15 seconds. Indeed, this time varying technique gives an appreciable improvement in both steady state error and maximum torque requirements, compared to when we use constant feedbacks K_p and K_v .

The second part of the simulation consisted in simulating the end effector motion of the three joint manipulator along an arbitrary preplanned path in joint space, which is given in table 4.4. The time varying technique, which provides a better performance, was used and the maximum allowable torque (force) was set to 5000 Nm (N). The required time of motion was set to 1.95 seconds. Simulation results, given by table 4.5, show a relatively high maximum error position and a final steady state error.

To summarize, one can say that with the computed torque technique adequate path tracking can be achieved only at the expense of high power actuators. Even though, the time varying technique lowers the torque requirements, relatively high torques are still needed for acceptable accuracy. This is mainly due to the fact that one tries to stabilize a nonlinear system with

Table 4.3. Simulation of joint motions from (1,0,-1) to (0.5,-0.6,-0.2) with time varying feedback gains.

THREE JOINT ROBOT MANIPULATOR-COMPUTED TORQUE TECHNIQUE
TIME VARYING FEEDBACK GAINS.

TIME	R	PHI	Z
0	1.000000	0	-1.000000
.050000	.853972	-.177880	-.731955
.100000	.564022	-.525944	-.266421
.150000	.504024	-.595975	-.195992
.200000	.504014	-.595986	-.195985
.250000	.504015	-.595985	-.195985
.300000	.504015	-.595985	-.195985

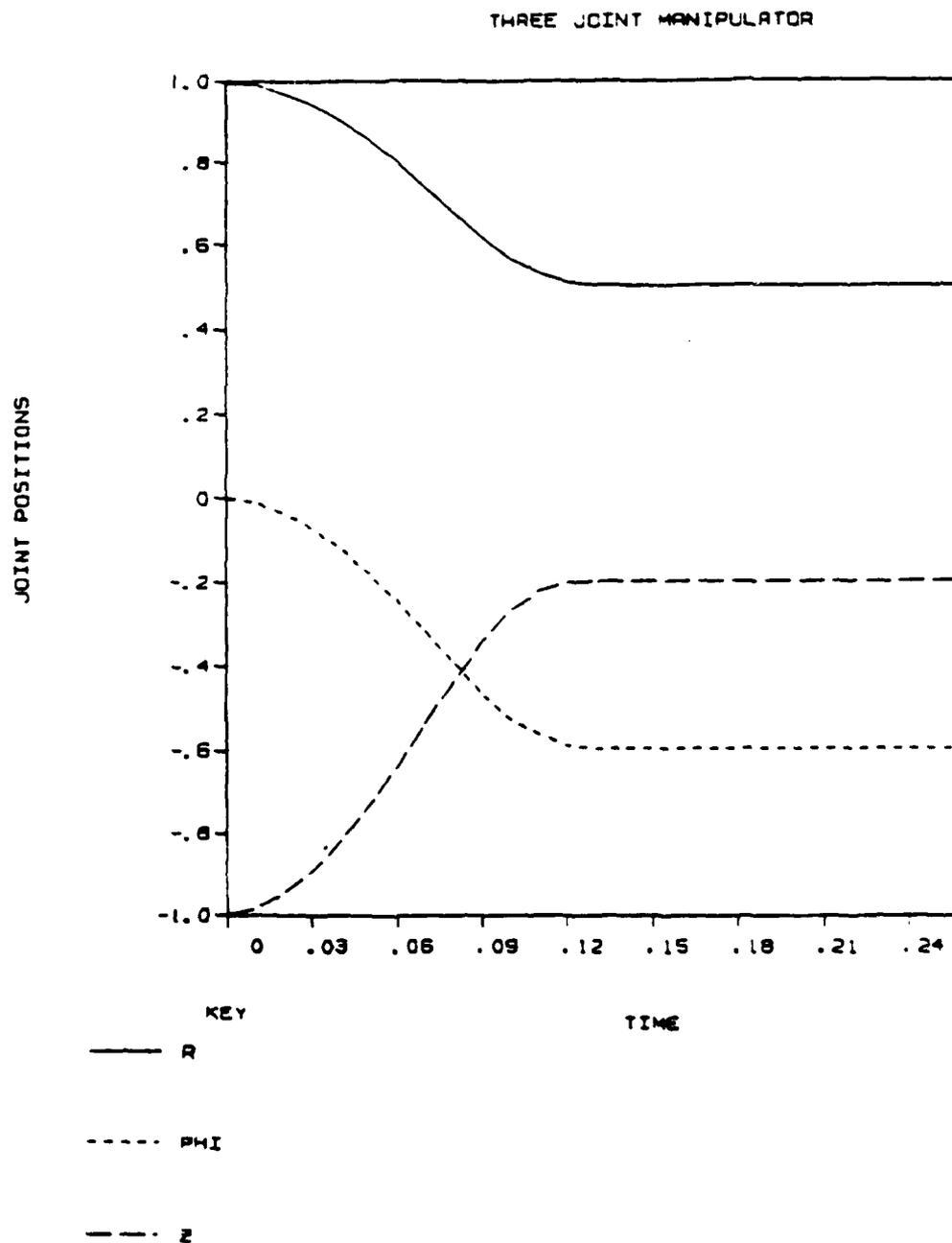


Figure 4.3. 'Computed torque' technique with time varying feedback gains. Simulation of joint motions from $(1, 0, -1)$ to $(0.5, -0.6, -0.2)$.

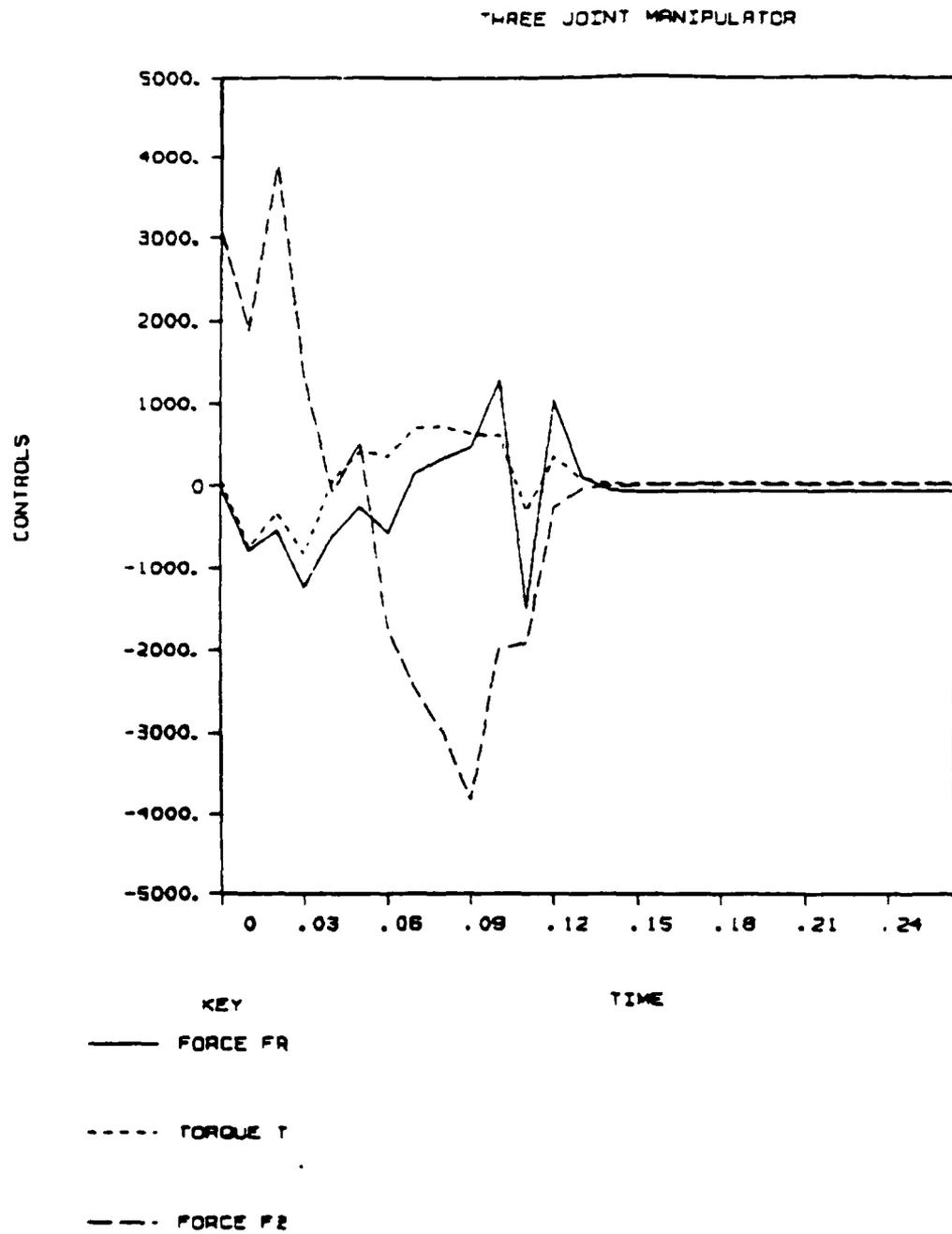


Figure 4.4. 'Computed torque' technique with time varying feedbacks. Controls.

THREE JOINT MANIPULATOR

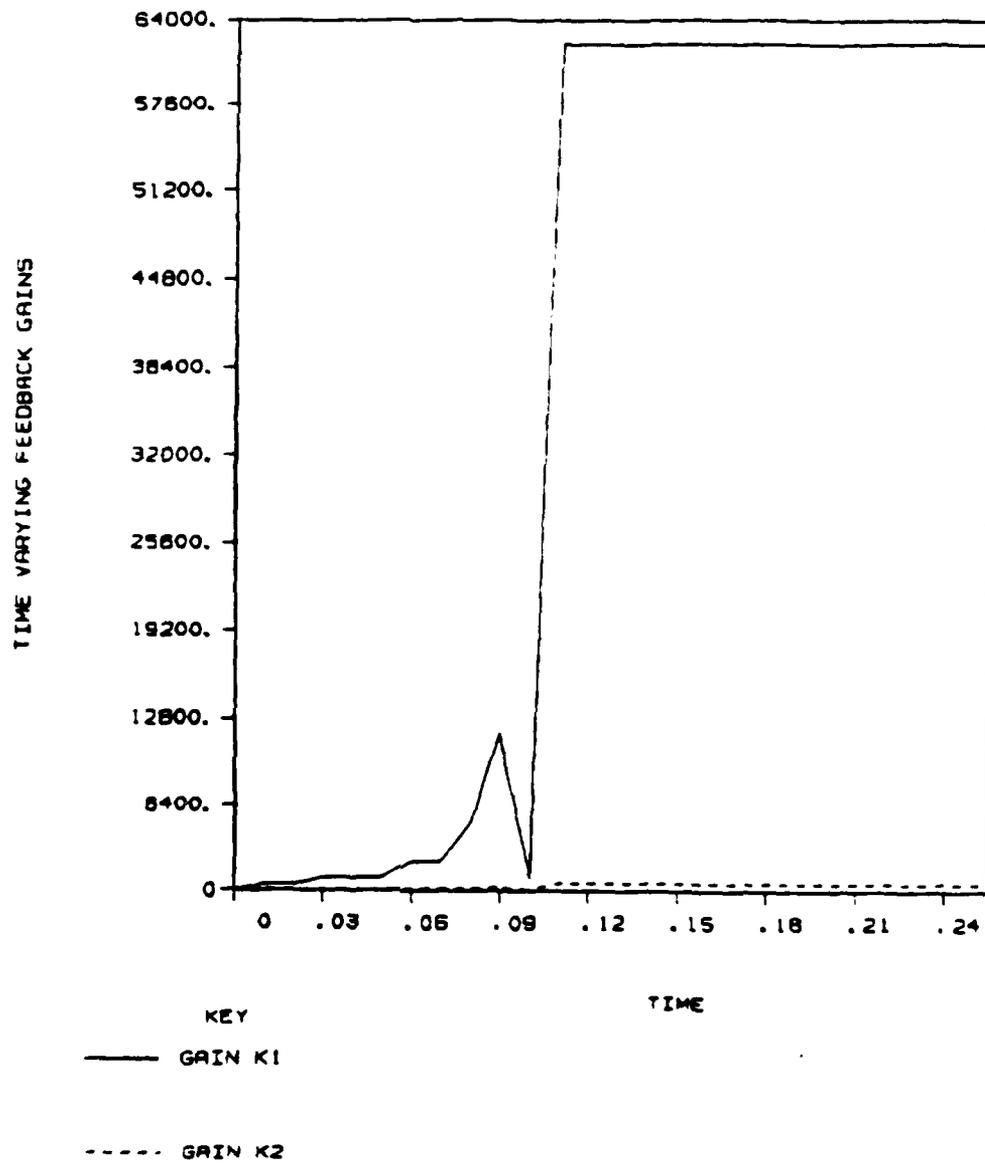


Figure 4.5. Time varying feedback gains, K_p and K_v .

Table 4.4. Preplanned path in joint space

JOINT POSITIONS						
r (m)	0.4	1	0.5	1.2	1.5	2
ϕ (rd)	0	0.5	-0.1	-0.9	-1.2	-1.8
z (m)	-1	-0.3	0.4	0.4	0.8	1.4

Table 4.5. 'Computed torque' technique with time varying feedback gains. Path tracking simulation.

Joint	Maximum tracking error	Final position error
Δr (mm)	11	4
$\Delta \phi$ (mrd)	10	4
Δz (mm)	11	4

basically a linear PD feedback law. In practice equations (3.2), on which the convergence process relies, are only approximately satisfied. Hence, it is expected that the tracking accuracy is worse than the one obtained by simulation.

We will comment more on this technique when comparing it to the feedback linearization which will be presented in the next section.

4.2 Feedback Linearization

The control of a n -joint robot manipulator is in fact the problem of controlling a dynamic system described by a set of n nonlinear differential equations. Due to the difficulty of the problem, one may attempt to linearize the system and design a linear control law.

A commonly used method to linearize a nonlinear system is the first order linearization (Taylor expansion). By this method; Golla, Gang and Hughes [5] linearized the dynamic equations and designed a linear state feedback controller. However as Vukobratovic [35] shows, when performance requirements (speed, accuracy, etc.) are raised, this approach does not lead to satisfactory results, mainly because at high speeds, higher order terms cannot be neglected.

A different approach to accomplish the linearization is via nonlinear feedbacks. This idea has been used in the past years and was the start of what has been termed 'geometric control theory'. Precisely, given a multi-input nonlinear system of the form:

$$\dot{x} = X^0(x) + \sum_{i=1}^m u_i(x) X^i(x)$$

where X^0, X^1, \dots, X^m are smooth vector fields on R^n (or a n -manifold M) and u_1, \dots, u_m are the controls. The question now is, under what conditions can one find a coordinate transformation and nonlinear feedbacks $u_i(x)$ ($i = 1, \dots, m$) such that the nonlinear system is linearizable. Lately, this question has been answered and a feedback linearization theorem with necessary and sufficient conditions has been proved [25].

In this section we will present this feedback linearization theorem and use it to derive a control algorithm with explicit feedbacks to control an n -joint robot manipulator.

4.2.1 Theoretical background

Lie algebra of vector fields [36] is extensively used in geometric control theory. In this section we give some definitions and theorems that will be used in later sections. We define the Kronecker indices and introduce the concept of nonlinear controllability.

Definition 4.2.1

The Lie product of two smooth vector fields $X(x)$ and $Y(x)$ is defined as:

$$[X, Y](x) = X_x(x)Y(x) - Y_x(x)X(x)$$

where X_x and Y_x are the Jacobians of X and Y .

Definition 4.2.2

Let f be a smooth real valued function of a manifold M^n onto R^1 .

The Lie operator L is defined as:

$$(L_X f)(x) = (Xf)(x) = \langle X(x), df(x) \rangle = \sum_i a_i(x) \frac{\partial f(x)}{\partial x_i}$$

where $X(x) = (a_1(x), \dots, a_n(x))^T$.

for notational convenience, we will use the following notations:

$$(\text{ad}X, Y) = [X, Y], \quad ((\text{ad}^0 X, Y) = Y)$$

$$(\text{ad}^2 X, Y) = [X, [X, Y]] = [X, (\text{ad}X, Y)]$$

$$(\text{ad}^k X, Y) = [X, (\text{ad}^{k-1} X, Y)]$$

Definition 4.2.3

Let f be a smooth real valued function of a manifold M^n onto R^1 .

In terms of operator notion, the Lie product $[X, Y]$ can also be defined as:

$$[X, Y]f = Y(x)(Xf)(x) - X(x)(Yf)(x).$$

Definition 4.2.4

Let G be a finite dimensional vector space. If the Lie product $[X, Y]$ defined on G ($X, Y \in G$) satisfies

1. $[\alpha X_1 + X_2, Y] = [\alpha X_1, Y] + [X_2, Y] = [X_1, \alpha Y] + [X_2, Y] = \alpha[X_1, Y] + [X_2, Y]$ for α real.
2. $[X, Y] = -[Y, X]$ (anticommutative)
3. $[[X, Y], Z] + [[Y, Z], X] + [[Z, X], Y] = 0$ (Jacobi Identity)

Then G is said to be a Lie algebra.

Definition 4.2.5

Given two Lie algebras G_1 and G_2 . Given also a one-to-one mapping ℓ from G_1 onto G_2 . Then ℓ is said to be a Lie algebra isomorphism if it satisfies:

1. For any real c_1, c_2 then

$$\ell(c_1 X_1 + c_2 X_2) = c_1 \ell(X_1) + c_2 \ell(X_2) = c_1 Y_1 + c_2 Y_2$$

2. $\ell[X_1, X_2] = [\ell(X_1), \ell(X_2)] = [Y_1, Y_2]$

where $X_1, X_2 \in G_1$ and $Y_1, Y_2 \in G_2$.

If ℓ is also differentiable, it is said to be a Lie algebra diffeomorphism.

Theorem 4.2.1

If ℓ is a Lie algebra diffeomorphism from a Lie algebra G_1 onto a Lie algebra G_2 , then the Jacobian ℓ_* of ℓ is also a Lie algebra diffeomorphism from G_1 onto G_2 with $\ell_*[X, Y] = [\ell_* X, \ell_* Y]$.

Definition 4.2.6: Involutiveness

Let $C = \{X^1, \dots, X^k\}$ be a set of smooth vector fields on R^n (or M^n), with $X^i(P), \dots, X^k(P)$ linearly independent for some point P . Then the set C is said to be involutive, if for any $X^i, X^j \in C$ there exist smooth, real valued functions $a_1^{ij}, \dots, a_k^{ij}$ such that the Lie product $[X^i, X^j](x) = \sum_{m=1}^k a_m^{ij}(x) X^m(x)$, x in the neighborhood of P , i.e. $[X, Y](x) \in \text{span of } C$.

Definition 4.2.7. Kronecker indices

The Kronecker indices for a matrix pair (A, B) are defined as follows:

$$\text{Let } R^i = [B, AB, \dots, A^{i-1}B]$$

$$\text{Let } \ell_1 = \dim R^1 = \dim B$$

$$\text{and } \ell_i = \dim R^i - \dim R^{i-1}$$

For an integer j , the Kronecker index K_j equals the total number of ℓ_i which are greater than or equal to j , i.e.

$$K_j = \{\# \ell_i \mid \ell_i \geq j\}$$

Example 4.2.1

$$\text{Let } A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad \text{Hence } AB = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\ell_1 = \dim B = 2$$

$$R^2 = (B, AB), \quad \dim R^2 = 4$$

$$\ell_2 = \dim R^2 - \dim R^1 = 4 - 2 = 2$$

The Kronecker indices are:

$$K_1 = \{\# \ell_i \mid \ell_i \geq 1\} = 2.$$

$$K_2 = \{\# \ell_i \mid \ell_i \geq 2\} = 2.$$

$$K_i = 0 \quad \text{for } i \geq 3.$$

Definition 4.2.8. Local controllability of a nonlinear system

Given the multi-input nonlinear system of the form

$$\dot{x} = X^0(x) + \sum_{i=1}^m u_i(x) X^i(x) \quad (4.1)$$

where X^0, X^1, \dots, X^m are smooth vector fields on M^n . Let P be the equilibrium point (corresponding to $u_i = 0$) so $X^0(P) = 0$. Then system (4.1) is said to be locally controllable about P , if for any time $t > 0$ there exists a control u such that any point in a full neighborhood R^n of P can be reached in time t by solutions initiating from P .

Theorem 4.2.2

A first order, sufficient test for local controllability of system (4.1) along its equilibrium point P is:

$$\dim \text{span}\{(\text{ad}^j X^0, X^i)(P) : i = 1, \dots, m; j = 0, 1, \dots\} = n$$

For a single input nonlinear system, this condition reduces to:

$$\dim \text{span}\{(\text{ad}^j X^0, X^1), j = 0, \dots, n-1\} \text{ being linearly independent.}$$

The proof can be found in [37].

4.2.2 Feedback linearization - single input case

In this section we define precisely the concept of feedback linearization and state the single input feedback linearization theorem.

Given the single input nonlinear system of the form:

$$\dot{x} = X^0(x) + u(x) X^1(x) \quad (4.2)$$

where X^0, X^1 are smooth vector fields on R^n (or M^n), u is the control and P is the point along which we want to linearize.

Given also the linear time invariant canonical form:

$$\dot{y} = Ay + bu, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & & & & \\ 0 & \dots & & 0 & 1 & \\ 0 & \dots & & & & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (4.3)$$

where u is the control.

Let $V^0(y) = Ay$ and $V^1(y) = b$, be vector fields on R^n , i.e.:

$$\dot{y} = V^0 + V^1 u \quad (4.4)$$

The feedback linearization problem can be expressed as follows:

When can one choose a coordinate transformation

$$y = \phi(x), \quad (4.5)$$

where ϕ is a local diffeomorphism (differentiable isomorphism), with $\phi(P) = 0$, $\phi_*(P)$ nonsingular and a nonlinear feedback control

$$u(x) = v(x) + \omega(x)\mu \quad (4.6)$$

where μ is a free new control, such that (4.5) and (4.6) transform the nonlinear system (4.2) into the linear system (4.3).

Specifically, with feedback alone, system (4.2) transforms to:

$$\dot{x} = X^0(x) + (v(x) + \omega(x)\mu)X^1(x)$$

$$\text{i.e.,} \quad \dot{x} = W^0(x) + \mu W^1(x) \quad (4.7)$$

where $W^0(x) = X^0(x) + v(x)X^1(x)$ and $W^1(x) = \omega(x)X^1(x)$ are smooth vector fields.

While with both coordinate change and feedback, we have:

$$\dot{y} = \phi_*(x)\dot{x}, \quad \phi_* \text{ is the Jacobian of } \phi.$$

$$\dot{y} = \phi_*(x)(X^0(x) + (v(x) + \omega(x)\mu)X^1(x))$$

$$\text{i.e.:} \quad \dot{y} = Y^0(y) + \mu Y^1(y) \quad (4.8)$$

where $Y^0(y) = \phi_*(x)(X^0(x) + v(x)X^1(x)) = \phi_*(x)W^0(x)$, which we want to be $V^0(y)$,

$$Y^1(y) = \phi_*(x)(\omega(x)X^1(x)) = \phi_*(x)W^1(x),$$

which we want to be $V^1(y)$.

We note that V^0 and V^1 generate a Lie algebra $L(V^0, V^1)$.

Suppose that system (4.8) is the linear canonical form (4.3).

Since ϕ is a diffeomorphism, then by theorem (4.1.1), ϕ_* is a Lie algebra diffeomorphism too, i.e. ϕ_* cannot change the structure of a Lie Algebra. Thus the Lie algebra $L(W^0, W^1)$ generated by W^0 and W^1 must be isomorphic to the Lie algebra $L(V^0, V^1)$. In other terms, it is necessary to be able to choose $v(x)$ and $\omega(x)$ to have such W^0 and W^1 .

Next we introduce the concept of feedback equivalence of two systems.

Definition (4.2.1)

Two systems are said to be feedback equivalent if one can be transformed into the other via a local coordinate change and a feedback.

Theorem 4.2.3

The n -dimensional, linear time invariant systems $\dot{x} = Ax + Bu$ and $\dot{y} = Cy + Du$ where A, C are $n \times n$ and B, D are $n \times m$ matrices; are feedback equivalent if and only if the pairs (A, B) and (C, D) have the same Kronecker indices.

Furthermore if (A, B) has Kronecker indices $K_1 \geq K_2 \geq \dots \geq K_m$, then it is feedback equivalent to the following canonical form, $\dot{z} = A^C z + B^C \mu$, μ input vector, and A^C is the Jordan block diagonal matrix given by

$$A^C = \text{diag}(A_1, \dots, A_m), \quad A_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & \\ & & & & 1 \\ 0 & \dots & & & 0 \end{bmatrix}$$

A_i being a $(K_i \times K_i)$ matrix with ones in the superdiagonal and zeros elsewhere. $B^C = (b^1, \dots, b^m)$, $b^i = [0 \dots 0 \ 1 \ 0 \dots 0]^t$ where b^i is a $(n \times 1)$ vector with 1 in the $\left(\sum_{j=1}^i K_j\right)$ component and zeros elsewhere.

The proof can be found in [38].

The following theorem for the single input linearization problem gives necessary and sufficient conditions for the existence of the diffeomorphism ϕ and the feedback control u . It also gives the explicit form of ϕ and u .

Theorem 4.2.4 - Single input feedback linearization theorem

Given the single input nonlinear system:

$$\dot{x} = X^0(x) + u(x)X^1(x) \quad (4.9)$$

where X^0, X^1 are smooth vector fields on M^n , u is the control and P is the point about which the linearization is desired.

$$\text{Let } \dot{y} = Ay + bu \quad (4.10)$$

$$\text{with } A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & & \\ 0 & & & & & 1 \\ 0 & \dots & & & & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

be a canonical linear time invariant system with control u .

Then three necessary and sufficient conditions for system (4.9) to be feedback equivalent to the canonical form (4.10) are:

i) there exist a real α such that:

$$X^0(P) + \alpha X^1(P) = 0.$$

ii) $X^1(P), (\text{ad}X^0, X^1)(P), \dots, (\text{ad}^{n-1}X^0, X^1)(P)$ are linearly independent (sufficient local controllability test, theorem 4.2.2).

iii) the set $\{X^1, (\text{ad}X^0, X^1), \dots, (\text{ad}^{n-2}X^0, X^1)\}$ is involutive.

Furthermore, when the above three conditions are satisfied and since the involutive condition 3 implies that there exist a smooth function $h: R^n \rightarrow R^1$ such that $h(P) = 0$, $(dh)(P) \neq 0$ and

$(\text{ad}^j X^0, X^1)h = \langle (\text{ad}^j X^0, X^1), dh \rangle = 0$ for $j = 0, 1, \dots, n-2$. Then the diffeomorphic coordinate change $y = \phi(x)$ with $\phi(P) = 0$ and $\phi_*(P)$ non-singular is given by:

$$y_j = (L_{X^0}^{j-1} h)(x) ; \quad j = 1, \dots, n$$

and the feedback $u(x) = v(x) + \omega(x)\mu$, where μ is a new free control (of (4.10)) is given by:

$$v(x) = - \frac{(L_{X^0}^n h)(x)}{(L_{X^1} L_{X^0}^{n-1} h)(x)}$$

$$\omega(x) = \frac{1}{(L_{X^1} L_{X^0}^{n-1} h)(x)}$$

Proof.

The proof of this theorem is constructive, i.e. in the process of proving the three necessary and sufficient conditions, the coordinate transform and the feedback u are constructed.

Necessary part

i) We have $V^0(y) = Ay$, so $V^0(0) = 0$. Since ϕ_* is a Lie algebra diffeomorphism, it is necessary that $W^0(P) = X^0(P) + v(P)X^1(P)$ be zero. Hence it is necessary that there exist a real α such that $X^0(P) + \alpha X^1(P) = 0$.

ii) We note that $v^1(0), (\text{ad} V^0, v^1)(0), \dots, (\text{ad}^{n-1} V^0, v^1)(0)$ are linearly independent (i.e., the linear controllability condition is satisfied). Again, since ϕ_* is a Lie algebra isomorphism, it is necessary that $W^1(P), \dots, (\text{ad}^{n-1} W^0, W^1)(P)$ are linearly independent since $L(W^0, W^1)$ must be isomorphic to

$L(V^0, V^1)$. But, $(\text{ad}W^0, W^1) = [X^0 + vX^1, \omega X^1] = [X^0, \omega X^1] + [vX^1, \omega X^1]$
 $= \omega[X^0, X^1] + (X^0\omega)X^1 + v(X^1\omega)X^1 - \omega(X^1v)X^1$, and since $W^1 = \omega X^1$,
 then $W^1(P)$, $[W^0, W^1](P)$ are independent if and only if $X^1(P)$,
 $[X^0, X^1](P)$ are independent.

Continuing with (ad^2W^0, W^1) etc., we obtain the second necessary condition:

$$X^1(P), (\text{ad}X^0, X^1)(P), \dots, (\text{ad}^{n-1}X^0, X^1)(P)$$

are linearly independent. This condition is sufficient for system (4.9) to be locally controllable at P . However we can have system (4.9) locally controllable at P but not have the second condition of the theorem satisfied. In this case we cannot transform via feedback and a coordinate change to the linear system (4.10).

(iii) At this point, one should note that for the linear system (4.10), $(\text{ad}^nV^0, V^1) = 0$. Also $[(\text{ad}^jV^0, V^1), (\text{ad}^\ell V^0, V^1)] = 0$ for all j, ℓ . Again, we must require this for the similar products of W^0, W^1 . Indeed, computing in the linear system shows $\{V^1, (\text{ad}V^0, V^1), \dots, (\text{ad}^{n-2}V^0, V^1)\}$ is an involutive set. Thus again since the Lie algebras $L(W^0, W^1), L(V^0, V^1)$ are isomorphic, we require that $\{W^1, \dots, (\text{ad}^{n-2}W^0, W^1)\}$ be involutive. But, $(\text{ad}W^0, W^1) = \omega[X^0, X^1] + (X^0\omega)X^1 + v(X^1\omega)X^1 - \omega(X^1v)X^1$. So as for the second necessary condition, we conclude that it is necessary that:

$$\{X^1, \dots, (\text{ad}^{n-2}X^0, X^1)\} \text{ is involutive.}$$

Sufficient part.

We want to show that when the three conditions are satisfied, there exist a diffeomorphism ϕ and a feedback u such that systems (4.9) and (4.10) are feedback equivalent. We will also give the explicit form of ϕ and u .

By necessary condition 1, there is an α such that $X^0(P) + \alpha X^1(P) = 0$. One can rewrite system (4.9) as: $\dot{x} = (X^0(x) + \alpha X^1(x)) + (u(x) - \alpha)X^1(x)$, i.e. $\dot{x} = \bar{X}^0(x) + \bar{u}(x)X^1(x)$, where $\bar{X}^0(x) = X^0(x) + \alpha X^1(x)$ vanishes at P and $\bar{u}(x) = (u(x) - \alpha)$ is a new control. Thus we can and will assume that $X^0(P) = 0$.

The involutive necessary condition 3 implies there exists a smooth function $h: \mathbb{R}^n \rightarrow \mathbb{R}^1$, with $(dh)(x) \neq 0$, such that:

$$(\text{ad}^j_{X^0}, X^1)h = 0 \quad j = 0, \dots, n-2. \quad (4.11)$$

We will also choose h so:

$$h(P) = 0, \quad (dh)(P) \neq 0.$$

Now we let the coordinate change ϕ be:

$$y_j = (L_{X^0}^{j-1} h)(x) \quad j = 1, \dots, n \quad (4.12)$$

with $y = (y_1, \dots, y_n) = \phi(x)$.

We claim that the map ϕ satisfies $\phi(P) = 0$ and $\phi_*(P)$ is non-singular, i.e. ϕ is a local diffeomorphism.

a) Indeed we have already chosen h such that $h(P) = 0$ i.e., $y_1(P) = 0$. Also, $y_2(P) = (L_{X^0} h)(P) = \langle X^0(P), dh(P) \rangle = 0$, since $X^0(P) = 0$. Similarly $y_j(P) = (L_{X^0}^{j-1} h)(P) = 0$ $j = 2, \dots, n$; because $X^0(P) = 0$. Hence $\phi(P) = 0$.

b) We wish to show that $\phi_*(P)$ is non-singular, i.e. $dh(P)$, $(dL_{X^0} h)(P), \dots, (dL_{X^0}^{n-1} h)(P)$ are linearly independent. We already have $dh(P) \neq 0$. Suppose $(dh)(P)$ and $(dL_{X^0} h)(P)$ are dependent, i.e. $\exists \alpha \neq 0$ such that

$$(dL_{X^0} h)(P) = \alpha(dh)(P).$$

Then, $(ad_{X^0, X^1} h)(P) = [X^0, X^1]h(P) = X^1(X^0 h)(P) - X^0(X^1 h)(P) = X^1(X^0 h)(P) = \langle X^1, dL_{X^0} h \rangle(P) = \langle X^1, \alpha dh \rangle(P) = \alpha \langle X^1, dh \rangle(P)$ which means that $X^1(P)$ and $(ad_{X^0, X^1} h)(P)$ are dependent. Inductively if $(dL_{X^0}^j h)(P)$ is a linear combination of $(dL_{X^0}^i h)(P)$ $i = 0, \dots, j-1$; we conclude that $(ad_{X^0, X^1}^{j+1} h)(P)$ is a linear combination of $(ad_{X^0, X^1}^i h)(P)$ $0 \leq i \leq j$, which contradicts the independence of $X^1(P), \dots, (ad_{X^0, X^1}^{n-1} h)(P)$ of necessary condition 2.

Hence $\phi_*(P)$ is non-singular and ϕ is a local diffeomorphism given by (4.12). Next, we construct the feedback u .

To shorten notation, let

$$V(x) = X^0(x) + u(x)X^1(x) = \dot{x}$$

First we wish to show that

$$(L_V^j h)(x) = (L_{X^0}^j h)(x), \quad j = 1, \dots, n-1 \quad (4.13)$$

and

$$L_{X^1} L_{X^0}^j h = 0 \quad j = 1, \dots, n-2 \quad (4.14)$$

For $j = 1$

$$L_V h = (X^0 + uX^1)h = X^0 h = L_{X^0} h$$

For $j = 2$

$$L_V^2 h = L_V L_V h = (X^0 + uX^1) L_{X^0} h = L_{X^0}^2 h + u L_{X^1} L_{X^0} h$$

Again from (4.11)

$$(\text{ad} X^0, X^1)h = X^1(X^0 h) - X^0(X^1 h) = X^1(X^0 h) = L_{X^1} L_{X^0} h = 0$$

Hence

$$L_V^2 h = L_{X^0}^2 h,$$

Continuing in this fashion and as long as $(\text{ad}^{j-1} X^0, X^1)h = 0$,

which is valid for $1 \leq j \leq n-1$ we have

$$L_V^j h = L_{X^0}^j h \quad 1 \leq j \leq n-1$$

Since $X^1(P), \dots, (\text{ad}^{n-1} X^0, X^1)(P)$ are independent and $X^0(P) = 0$

it follows that $X^1(P) \neq 0$. We have $(\text{ad} X^0, X^1)h = X^1(X^0 h) -$

$X^0(X^1 h) = X^1(X^0 h) = L_{X^1} L_{X^0} h = 0$. Inductively, from

$(\text{ad}^j X^0, X^1)h = 0$ for $j = 1, \dots, n-2$ we have

$$L_{X^1} L_{X^0}^j h = 0 \quad j = 1, \dots, n-2$$

The feedback u is obtained as follows. Differentiating the coordinate transformation (4.12) we have

$$\begin{aligned} \dot{y}_1 &= (dh)(x) \cdot \dot{x} = \langle X^0 + uX^1, dh \rangle = L_V h = L_{X^0} h = y_2 \\ \dot{y}_2 &= L_V^2 h = L_{X^0}^2 h = y_3 \\ &\vdots \\ \dot{y}_{n-1} &= L_V^{n-1} h = L_{X^0}^{n-1} h = y_n \end{aligned} \tag{4.15}$$

$$\begin{aligned} \dot{y}_n &= (dL_{X^0}^{n-1} h)(x) \cdot \dot{x} = \langle X^0 + uX^1, dL_{X^0}^{n-1} h \rangle \\ &= (L_{X^0}^n h)(x) + u(x) (L_{X^1} L_{X^0}^{n-1} h)(x) \end{aligned}$$

But $u(x) = v(x) + \omega(x)u$. Since we want system (4.15) to be the linear canonical form (4.10), we have

$$\dot{y}_n = (L_{X^0}^n h)(x) + v(x) (L_{X^1} L_{X^0}^{n-1} h)(x) + \omega(x) (L_{X^1} L_{X^0}^{n-1} h)(x)u$$

One can then choose

$$v(x) = - \frac{(L_{X^0}^n h)(x)}{(L_{X^1} L_{X^0}^{n-1} h)(x)}$$

$$\omega(x) = \frac{1}{(L_{X^1} L_{X^0}^{n-1} h)(x)}$$

to make the last equation become $\dot{y}_n = u$, and system (4.15) is the linear system (4.10).

Remark.

In many applications the linearization is to be done around the equilibrium point corresponding to zero controls. In this case P is such that $X^0(P) = 0$ and the first condition of the theorem becomes trivial ($\alpha = 0$).

4.2.3 Feedback equivalence - multi-input case

In this part we give the multi-input feedback linearization theorem which is an extension of the single input one. We will show that when the linearization works, the nonlinear system is feedback equivalent to a canonical linear system determined by the Kronecker indices of the nonlinear system.

Theorem 4.2.5. Multi-input feedback linearization theorem

Consider the multi-input nonlinear system of the form:

$$\dot{x} = X^0(x) + \sum_{i=1}^m u_i(x) X^i(x) \quad (4.16)$$

Let P be the point around which we wish to linearize.

The necessary and sufficient conditions for system (4.16), with Kronecker indices $K_1 \geq K_2 \geq \dots \geq K_m$, to be feedback equivalent to the linear time invariant canonical form $\dot{z} = A^c z + B^c u$ as given by theorem (4.2.3) are:

1 - There exist reals $\alpha_1, \dots, \alpha_m$ so

$$X^0(P) + \sum_{i=1}^m \alpha_i X^i(P) = 0$$

2 - Let $C(x) = \{X^1, \dots, (\text{ad}^{K_1-1} X^0, X^1), X^2, \dots, (\text{ad}^{K_2-1} X^0, X^2), \dots, X^m, \dots, (\text{ad}^{K_m-1} X^0, X^m)\}$. Then $\dim C(P) = n$ (local controllability test).

3 - For each $i = 1, \dots, m$ let

$$C_i = \{X^1, \dots, (\text{ad}^{K_i-2} X^0, X^1), X^2, \dots, (\text{ad}^{K_i-2} X^0, X^2), \dots, X^m, \dots, (\text{ad}^{K_i-2} X^0, X^m)\}$$

Then C_i must be involutive and $\text{Span } C_i(x) =$

$\text{Span } (C_i \cap C)(x)$ i.e., $C_i(x) \subset C(x)$ for all i .

The proof of this theorem is similar to the one given for the single input case and can be found in [25].

Remark:

As for the single input theorem when the point of interest P is the equilibrium point (in most applications it is) such that $X^0(P) = 0$, the first condition of the theorem becomes trivial ($\alpha_i = 0$, for all i).

In the single input case the construction of the diffeomorphism ϕ and the feedback control u from the real valued function h was proved. But for the multi-input case the construction work has not been proved, we rather guess the functions $h_i(x)$ as it will be shown for the n -joint robot manipulator.

A special case of interest is when the Kronecker indices are all less or equal to two. In this case the third condition is always true. Indeed, $C_i = \{X^1, X^2, \dots, X^m\}$ which is an involutive set and $C_i \subset C$ for all i .

4.2.4 Application to robotics

In this section we will design a nonlinear controller for robot manipulators by employing the feedback linearization technique. We will first show that the feedback linearization theorem is applicable to linearize the dynamic equations of a n -joint robot manipulator. Then, we will derive a general expression of the feedback controls. We will illustrate this design procedure by two examples, a two joint and three joint manipulator. Computer simulations for performance evaluation are also given.

In chapter III we have seen that the dynamic equations of a n -joint manipulator, as given by (3.1), have the following form.

$$D(q)\ddot{q} + H(q, \dot{q}) + G(q) = \tau \quad (4.17)$$

where $q = (q_1, \dots, q_n)$ is a vector of the actual joint positions.

If we let

$$q_1 = x_{2i-1}$$

$$\dot{q}_1 = x_{2i} \quad i = 1, \dots, n$$

then, the state space representation of (4.17) can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \\ \dot{x}_{2i-1} \\ \dot{x}_{2i} \\ \vdots \\ \dot{x}_{2n-1} \\ \dot{x}_{2n} \end{bmatrix} = \begin{bmatrix} x_2 \\ f_1(x) \\ x_3 \\ f_2(x) \\ \vdots \\ x_{2i} \\ f_i(x) \\ \vdots \\ x_{2n} \\ f_n(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_1(x) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_2(x) + \dots + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_i(x) + \dots$$

$$+ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_n(x) \quad (4.18)$$

where $u_i(x)$ contains a nonlinear combination of some torques τ_j .

We will assume that $f_i(x)$ is a smooth real function, for all i .

System (4.18) is then a multi-input nonlinear system of the form:

$$\dot{x} = X^0(x) + \sum_{i=1}^n X^i(x) u_i(x)$$

where $X^0: X^i, i = 1, \dots, n$ are smooth vector fields as given by (4.18).

Let P be the equilibrium point of (4.18), corresponding to $u_i = 0$, i.e.

$$X^0(P) = 0.$$

Next we show that this state representation leads to Kronecker indices all less than or equal to 2.

i) Kronecker indices:

$$R^1 = \text{span} [X^1(P), \dots, X^n(P)] = \text{span} \begin{bmatrix} 0 & 0 \\ 1 & \vdots \\ 0 & \dots \\ \vdots & 0 \\ 0 & 1 \end{bmatrix}$$

$$l_1 = \dim R^1 = n.$$

$$R^2 = \text{span} [X^1(P), (\text{ad}X^0, X^1)(P), \dots, X^n(P), (\text{ad}X^0, X^n)(P)]$$

$$X_x^0(x) = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & & & \frac{\partial f_1}{\partial x_{2n}} \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & & & \frac{\partial f_2}{\partial x_{2n}} \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & \dots & & 0 & 1 \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & & & \frac{\partial f_n}{\partial x_{2n}} \end{bmatrix}$$

$$(\text{ad}X^0, X^1)(P) = X_x^0(x) \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} (P) = \begin{bmatrix} 1 \\ \frac{\partial f_1}{\partial x_2}(P) \\ 0 \\ \vdots \\ 0 \\ \frac{\partial f_n}{\partial x_2}(P) \end{bmatrix}$$

$$(\text{ad}X^0, X^2)(P) = X_x^0(x) \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} (P) = \begin{bmatrix} 0 \\ \frac{\partial f_1}{\partial x_4}(P) \\ 1 \\ \frac{\partial f_2}{\partial x_4}(P) \\ \vdots \\ 0 \\ \frac{\partial f_n}{\partial x_4}(P) \end{bmatrix}$$

$$(\text{ad}X^0, X^n)(P) = X_x^0(x) \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial f_1}{\partial x_{2n}}(P) \\ 0 \\ \vdots \\ \frac{\partial f_n}{\partial x_{2n}}(P) \end{bmatrix}$$

then,

$$R^2 = \text{span} \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ \frac{\partial f_1}{\partial x_2}(P) \\ 0 \\ \frac{\partial f_2}{\partial x_2}(P) \\ \vdots \\ 0 \\ \frac{\partial f_n}{\partial x_2}(P) \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{\partial f_1}{\partial x_4}(P) \\ 1 \\ \frac{\partial f_2}{\partial x_4}(P) \\ \vdots \\ 0 \\ \frac{\partial f_n}{\partial x_4}(P) \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{\partial f_1}{\partial x_{2n}}(P) \\ 0 \\ \frac{\partial f_2}{\partial x_{2n}}(P) \\ \vdots \\ 1 \\ \frac{\partial f_n}{\partial x_{2n}}(P) \end{bmatrix} \right\}$$

Hence,

$$\dim R^2 = 2n \quad (\det = -1)$$

$$l_2 = \dim R^2 - \dim R^1 = n$$

$$l_i = 0 \quad \text{for } i \geq 3 .$$

The Kronecker indices are then:

$$K_1 = \{\# l_i \mid l_i \geq 1\} = 2$$

$$K_2 = \{\# l_i \mid l_i \geq 2\} = 2$$

$$\vdots$$

$$K_n = \{\# l_i \mid l_i \geq n\} = 2$$

$$K_i = 0 \quad \text{for } i > n .$$

Hence all Kronecker indices are all less than or equal to 2. Next we check the three necessary and sufficient conditions of the feedback linearization theorem.

Since we are interested in linearizing around the equilibrium point P , with $X^0(P) = 0$, the first condition is trivial.

Also, due to the fact that all Kronecker indices are less than or equal to 2, the third condition is immediate.

For the second condition let

$$C(x) = \{X^1, (\text{ad}X^0, X^1), \dots, X^n, (\text{ad}X^0, X^n)\}$$

$$\dim C(P) = \dim R^2 = 2n$$

Hence, the nonlinear system (4.18) is locally controllable about P .

Thus, the feedback linearization theorem is applicable and the nonlinear system (4.18), which is the representation of a robot manipulator dynamic equations, is feedback equivalent to the following linear time invariant canonical system:

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & & \vdots \\ 0 & 0 & 0 & 1 & & \\ 0 & 0 & 0 & 0 & \ddots & \\ \vdots & & & & \ddots & 1 \\ 0 & \dots & & & & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \mu_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mu_2 + \dots + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mu_n$$

which consists of n uncoupled second order subsystems of the form:

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \mu_1 \\ &\vdots \\ \dot{y}_{2n-1} &= y_{2n} \\ \dot{y}_{2n} &= \mu_n \end{aligned} \tag{4.19}$$

where μ_i 's are the controls.

ii) Synthesis of the feedback controls.

As mentioned previously, the construction of the coordinate change ϕ and the feedback control u is not proved for the multi-input feedback linearization theorem. Proceeding as for the single input case, we will 'guess' n real functions $h_1, \dots, h_n: \mathbb{R}^n \rightarrow \mathbb{R}^1$, such that

$$* h_i(P) = 0 \quad i = 1, \dots, n$$

$$* dh_1(P), \dots, dh_n(P) \text{ are linearly independent.}$$

$$* (\text{ad}^j X^0, X^i)h_m = 0 \quad \forall i, j, m \text{ i.e.:}$$

$$L_{X^i} h_j = \langle X^i, dh_j \rangle = 0 \quad \forall i, j.$$

Choosing h_1, \dots, h_n to meet these three conditions can be done by inspecting the vector fields X^0, \dots, X^n as given by (4.18).

One may choose

$$h_1(x) = x_1$$

$$h_2(x) = x_3$$

$$\vdots$$

$$h_i(x) = x_{2i-1}$$

$$\vdots$$

$$h_n(x) = x_{2n-1}$$

Indeed

$$* h_1(0) = \dots = h_n(0) = 0$$

$$* dh_1(0) = (1, 0, \dots, 0)^t, dh_2(0) = (0, 0, 1, \dots, 0)^t, \dots, dh_n(0) = (0, \dots, 1, 0)^t \text{ are linearly independent.}$$

$$* L_{x^i} h_j = \langle X^i, dh_j \rangle = 0 \quad \forall i, j.$$

The coordinate change is chosen as:

$$\begin{aligned} y_1 &= h_1(x) = x_1 \\ y_2 &= (L_{x^0} h_1)(x) = \langle X^0, dh_1 \rangle = \langle X^0, (1, 0, \dots, 0)^t \rangle = x_2 \\ y_3 &= h_2(x) = x_3 \\ y_4 &= (L_{x^0} h_2)(x) = \langle X^0, dh_2 \rangle = x_4 \\ &\vdots \\ y_{2i-1} &= h_i(x) = x_{2i-1} \\ y_{2i} &= (L_{x^0} h_i)(x) = \langle X^0, dh_i \rangle = x_{2i} \\ &\vdots \\ y_{2n-1} &= h_n(x) = x_{2n-1} \\ y_{2n} &= (L_{x^0} h_n)(x) = \langle X^0, dh_n \rangle = x_{2n} \end{aligned} \tag{4.20}$$

Equations (4.20) define the coordinate transform $y = \phi(x) = Ix$ ($I = \text{Identity}$), with $\phi(0) = 0$ and $\phi_*(0) = I$.

The coordinate transform is then the identity, i.e. no explicit coordinate change is needed and the linearization will be accomplished by feedback alone.

At this point we note that if we choose the state representation (4.18) such that $f_i(x) = 0$ for all i , we get exactly system (4.19) with controls u_i 's instead of u_i 's.

Since $\phi(x) = Ix$, obviously $\dot{y} = \dot{x}$. But for the purpose of showing the feedback synthesis for any diffeomorphism $\phi(x)$ we will continue the construction work as follows:

Differentiating (4.20) gives:

$$\begin{aligned}
 \dot{y}_1 &= dh_1 \cdot \dot{x} = \langle X^0 + u_1 X^1 + \dots + u_n X^n, dh_1 \rangle \\
 &= \langle X^0, dh_1 \rangle = L_{X^0} h_1 = y_2 \\
 \\
 \dot{y}_2 &= dL_{X^0} h_1 \cdot \dot{x} = \langle X^0 + u_1 X^1 + \dots + u_n X^n, dL_{X^0} h_1 \rangle \\
 &= L_{X^0}^2 h_1 + u_1 L_{X^1} L_{X^0} h_1 + \dots + u_n L_{X^n} L_{X^0} h_1 \\
 &\vdots \\
 \dot{y}_{2i-1} &= dh_i \cdot \dot{x} = \langle X^0 + u_1 X^1 + \dots + u_n X^n, dh_i \rangle \quad (4.21) \\
 &= \langle X^0, dh_i \rangle = L_{X^0} h_i = y_{2i} \\
 \\
 \dot{y}_{2i} &= dL_{X^0} h_i \cdot \dot{x} = \langle X^0 + u_1 X^1 + \dots + u_n X^n, dL_{X^0} h_i \rangle \\
 &= L_{X^0}^2 h_i + u_1 L_{X^1} L_{X^0} h_i + \dots + u_i L_{X^i} L_{X^0} h_i + \dots \\
 &\quad + u_n L_{X^n} L_{X^0} h_i \\
 &\vdots \\
 \dot{y}_{2n-1} &= dh_n \cdot \dot{x} = \langle X^0 + u_1 X^1 + \dots + u_n X^n, dh_n \rangle \\
 &= \langle X^0, dh_n \rangle = L_{X^0} h_n = y_{2n} \\
 \\
 \dot{y}_{2n} &= dL_{X^0} h_n \cdot \dot{x} = \langle X^0 + u_1 X^1 + \dots + u_n X^n, dL_{X^0} h_n \rangle \\
 &= L_{X^0}^2 h_n + u_1 L_{X^1} L_{X^0} h_n + \dots + u_n L_{X^n} L_{X^0} h_n
 \end{aligned}$$

with

$$L_{X^0} h_1 = \langle X^0, dh_1 \rangle = x_2$$

$$L_{X^0}^2 h_1 = \langle X^0, dL_{X^0} h_1 \rangle = \langle X^0, (0, 1, 0, \dots, 0)^t \rangle = f_1(x)$$

$$L_{X^1} L_{X^0} h_1 = \langle X^1, dL_{X^0} h_1 \rangle = \langle X^1, (0, 1, 0, \dots, 0)^t \rangle = 1$$

⋮

$$L_{X^i} L_{X^0} h_1 = \langle X^i, dL_{X^0} h_1 \rangle = \langle X^i, (0, 1, 0, \dots, 0)^t \rangle = 0 \quad i = 2, \dots, n$$

⋮

$$L_{X^0} h_i = \langle X^0, dh_i \rangle = \langle X^0, (0, \dots, 0, \underset{2i-1}{1}, 0, \dots, 0)^t \rangle = x_{2i}$$

$$L_{X^0}^2 h_i = \langle X^0, dL_{X^0} h_i \rangle = \langle X^0, (0, \dots, 0, \underset{2i}{1}, 0, \dots, 0)^t \rangle = f_i(x)$$

$$L_{X^i} L_{X^0} h_i = \langle X^i, dL_{X^0} h_i \rangle = \langle X^i, (0, \dots, 0, \underset{2i}{1}, 0, \dots, 0)^t \rangle = 1$$

$$L_{X^j} L_{X^0} h_i = 0 \quad \forall i \neq j \quad i = 1, \dots, n$$

⋮

$$L_{X^0} h_n = \langle X^0, dh_n \rangle = x_{2n}$$

$$L_{X^n} L_{X^0} h_n = \langle X^n, dL_{X^0} h_n \rangle = \langle X^n, (0, \dots, 0, 1)^t \rangle = 1$$

$$L_{X^i} L_{X^0} h_n = 0 \quad i = 1, \dots, n-1$$

Hence (4.21) becomes:

$$\left. \begin{aligned}
 \dot{y}_1 &= y_2 \\
 \dot{y}_2 &= f_1(x) + u_1(x) \\
 \dot{y}_3 &= y_4 \\
 \dot{y}_4 &= f_2(x) + u_2(x) \\
 &\vdots \\
 \dot{y}_{2i-1} &= y_{2i} \\
 \dot{y}_{2i} &= f_i(x) + u_i(x) \\
 &\vdots \\
 \dot{y}_{2n-1} &= y_{2n} \\
 \dot{y}_n &= f_n(x) + u_n(x)
 \end{aligned} \right\} \quad (4.22)$$

The nonlinear system (4.18) is then feedback equivalent to system (4.22). Now, we want to make system (4.22) be the linear canonical form (4.19). This is accomplished by letting

$$\dot{y}_{2i} = f_i(x) + u_i(x) = \mu_i$$

and the explicit feedbacks $u_i(x)$ are given by:

$$u_i(x) = -f_i(x) + \mu_i \quad (4.23)$$

where μ_i is a new free control.

The control problem of the robot manipulator is now reduced to controlling the n linear canonical and uncoupled subsystems (4.19).

The objective of the control is to move the robot hand from one path point to another desired one, or through the inverse kinematics, from one set of initial joint positions $x_1, x_3, \dots, x_{2n-1}$ to the desired joint positions w_1, w_3, \dots, w_{n-1} . Equivalently, via the coordinate transform (4.20), this is the problem of moving from the states $y_1, y_3, \dots, y_{2n-1}$ (of canonical form (4.19)) to desired states $z_1, z_3, \dots, z_{2n-1}$.

Subsystems of (4.19) are in this form:

$$\left\{ \begin{array}{l} \dot{y}_{2i-1} = y_{2i} \\ \dot{y}_{2i} = \mu_i \end{array} \right. \quad i = 1, \dots, n \quad (4.24)$$

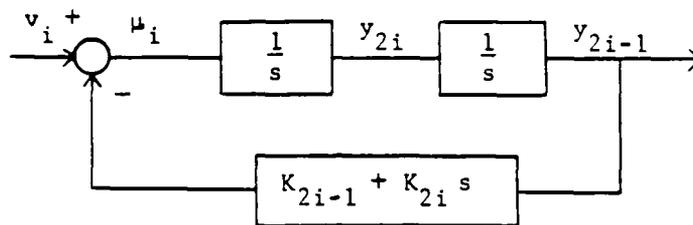
which is a double integrator, type two system with transfer function:

$$\frac{y_{2i-1}(s)}{\mu_i(s)} = \frac{1}{s^2}$$

System (4.24) can be stabilized by a conventional state feedback.

$$\text{Let } \mu_i = v_i - K_{2i-1}y_{2i-1} - K_{2i}y_{2i} \quad i = 1, \dots, n$$

i.e.:



with transfer function $\frac{y_{2i-1}(s)}{v_i(s)} = \frac{1}{s^2 + K_{2i}s + K_{2i-1}}$. To make the steady state response $y_{2i-1}(t)$ be the desired z_{2i-1} , let $v_i(s) = L_i \cdot z_{2i-1} \cdot u(s)$ where $u(s)$ is a step input. Hence,

$$\lim_{t \rightarrow \infty} \frac{y_{2i-1}(t)}{z_{2i-1}} = \lim_{s \rightarrow 0} s \frac{L_i \frac{1}{s}}{s^2 + K_{2i}s + K_{2i-1}} = \frac{L_i}{K_{2i-1}}$$

Hence $y_{2i-1}(\infty) = z_{2i-1}$ if $L_i = K_{2i-1}$. Therefore the control μ_i is:

$$\mu_i = K_{2i-1}(z_{2i-1} - y_{2i-1}) - K_{2i}y_{2i} \quad i = 1, \dots, n. \quad (4.25)$$

We have shown that the original system (4.18) and the linear canonical form (4.19) are feedback equivalent. This feedback equivalence implies that both systems have the same dynamic behavior. Again, in most robot applications, overshoot is to be avoided. To ensure the fastest response with no overshoot, the feedback gains K_{2i} , K_{2i-1} are chosen for a critically damped response of the second order subsystems (4.19).

By the coordinate transform (4.20) we can express u_i as:

$$u_i = K_{2i-1}(w_{2i-1} - x_{2i-1}) - K_{2i}x_{2i} \quad i = 1, \dots, n$$

Thus, the explicit feedbacks $u_i(x)$ are:

$$u_i(x) = -f_i(x) + K_{2i-1}(w_{2i-1} - x_{2i-1}) - K_{2i}x_{2i} \quad (4.26)$$

where $f_i(x)$ is given by (4.18)

K_{2i}, K_{2i-1} are feedback gains

w_{2i-1} is the desired joint position, $i = 1, \dots, n$

The actual control torques τ_i ($i = 1, \dots, n$) to be applied by the actuators are obtained by solving for τ_i , the set of equations $u_i(x)$; $i = 1, \dots, n$.

Note that we assume that the states x_1, \dots, x_{2n} (joint positions and velocities) are all measurable.

Here, we summarize the procedure we went through to derive the feedback controls.

First, we obtained the state space representation (4.18) from the dynamic equations of an n -joint robot manipulator. For a six-joint manipulator, the expressions of $u_i(x)$ and $f_i(x)$ can contain hundreds of algebraic terms. In spite of the complexity of the robot dynamic equations, state representation (4.18) led to Kronecker indices all less or equal to two. This made the multi-input feedback linearization theorem readily applicable. The coordinate transform was found to be the identity and, explicit nonlinear feedbacks were derived.

Next, we will illustrate this procedure by two examples, a two and three joint manipulator. For these two examples, a computer simulation study is conducted to evaluate the performance of the nonlinear controller. We will also compare with the 'computed torque' technique used in the previous section (4.1).

Example 4.2.2: Two joint manipulator

From the dynamic equations given by (3.6) and (3.7), one notices that due to the coupling effect, $\ddot{\theta}_1$ and $\ddot{\theta}_2$ are present in both equations. In order to derive the state space representation we first solve for $\ddot{\theta}_1$ and $\ddot{\theta}_2$. Then we obtain

$$\ddot{\theta}_1 = - \frac{a_2(x)b_1\ell_2}{a_2^2(x)-a_1(x)b_2\ell_2^2} (\ell_1\dot{\theta}_1^2\sin\theta_2 + g\cos(\theta_1+\theta_2)) \\ + \frac{b_2\ell_2^2}{a_2^2(x)-a_1(x)b_2\ell_2^2} (-b_1\ell_1\ell_2\sin\theta_2(2\dot{\theta}_1+\dot{\theta}_2)\dot{\theta}_2)$$

$$\begin{aligned}
& + b_3 g \ell_1 \cos \dot{\theta}_1 + b_1 g \ell_2 \cos(\dot{\theta}_1 + \dot{\theta}_2) + \frac{a_2(x) T_2 - b_2 \dot{\ell}_2^2 T_1}{a_2(x) - a_1(x) b_2 \dot{\ell}_2^2} \\
\ddot{\theta}_2 = & \frac{a_1(x) b_1 \dot{\ell}_2}{a_2(x) - a_1(x) b_2 \dot{\ell}_2^2} (\dot{\ell}_1 \dot{\ell}_1^2 \sin \dot{\theta}_2 + g \cos(\dot{\theta}_1 + \dot{\theta}_2)) \\
& + \frac{a_2(x)}{a_2(x) - a_1(x) b_2 \dot{\ell}_2^2} (b_1 \dot{\ell}_1 \dot{\ell}_2 \sin \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 \\
& - b_3 g \ell_1 \cos \dot{\theta}_1 - b_1 g \ell_2 \cos(\dot{\theta}_1 + \dot{\theta}_2)) + \frac{a_2(x) T_1 - a_1(x) T_2}{a_2(x) - a_1(x) b_2 \dot{\ell}_2^2}
\end{aligned} \tag{4.27}$$

where

$$\begin{aligned}
b_1 &= \frac{m_2}{2} + m_L \\
b_2 &= \frac{m_2}{4} + m_L \\
b_3 &= \frac{m_1}{2} + m_2 + m_L \\
b_4 &= \frac{m_1}{4} + m_2 + m_L
\end{aligned} \tag{4.28}$$

and

$$\begin{aligned}
a_1(x) &= b_4 \dot{\ell}_1^2 + b_2 \dot{\ell}_2^2 + 2b_1 \dot{\ell}_1 \dot{\ell}_2 \cos \dot{\theta}_2 \\
a_2(x) &= b_2 \dot{\ell}_2^2 + b_1 \dot{\ell}_1 \dot{\ell}_2 \cos \dot{\theta}_2
\end{aligned} \tag{4.29}$$

Let the state variables be:

$$\begin{aligned}
x_1 &= \dot{\theta}_1 \\
x_2 &= \dot{\theta}_2 \\
x_3 &= \dot{\theta}_2
\end{aligned} \tag{4.30}$$

$$\dot{x}_4 = \frac{1}{2}$$

Hence the state space representation is:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x) + u_1(x) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x) + u_2(x) \end{cases} \quad (4.31)$$

where

$$f_1(x) = -\frac{a_2(x)}{a_2^2(x) - a_1(x)b_2^2} b_1 l_1 l_2 x_2^2 \sin x_3 \quad (4.32)$$

$$g_1(x) = -\frac{a_2(x)}{a_2^2(x) - a_1(x)b_2^2} b_1 g l_2 \cos(x_1 + x_3) + \frac{b_2 l_2^2}{a_2^2(x) - a_1(x)b_2^2} (-b_1 l_1 l_2 \sin x_3 (2x_2 + x_4)x_4 + b_3 g l_1 \cos x_1 + b_1 g l_2 \cos(x_1 + x_3)) \quad (4.33)$$

$$u_1(x) = g_1(x) + \frac{a_2(x)T_2 - b_2 l_2^2 T_1}{a_2^2(x) - a_1(x)b_2^2} \quad (4.34)$$

$$f_2(x) = \frac{a_1(x)}{a_2^2(x) - a_1(x)b_2^2} b_1 l_1 l_2 x_2^2 \sin x_3 \quad (4.35)$$

$$g_2(x) = \frac{a_1(x)}{a_2^2(x) - a_1(x)b_2^2} b_1 g l_2 \cos(x_1 + x_3) + \frac{a_2(x)}{a_2^2(x) - a_1(x)b_2^2} (b_1 l_1 l_2 \sin x_3 (2x_2 + x_4)x_4 - b_3 g l_1 \cos x_1 - b_1 g l_2 \cos(x_1 + x_3)) \quad (4.36)$$

$$u_2(x) = g_2(x) + \frac{a_2(x)T_1 - a_1(x)T_2}{a_2^2(x) - a_1(x)b_2\ell_2^2} \quad (4.37)$$

System (4.31) can be put in this form:

$$\dot{x} = X^0(x) + X^1(x)u_1(x) + X^2(x)u_2(x) \quad (4.38)$$

where

$$X^0(x) = \begin{bmatrix} x_2 \\ f_1(x) \\ x_4 \\ f_2(x) \end{bmatrix}, \quad X^1(x) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad X^2(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.39)$$

$u_1(x)$, $u_2(x)$ are the controls, and $P = 0$ is the equilibrium point of (4.38).

We first check that indeed the Kronecker indices are all less or equal to 2.

$$\ell_1 = \dim R^1 = \dim \{X^1, X^2\} = 2$$

$$R^2 = \{X^1(P), (\text{ad}X^0, X^1)(P), X^2(P), (\text{ad}X^0, X^2)(P)\}$$

$$\text{ad}(X^0, X^1)(P) = \begin{bmatrix} 1 \\ -\frac{a_2^2 b_1 \ell_1 \ell_2 x_2 \sin x_3}{a_2^2 - a_1 b_2 \ell_2^2} \\ 0 \\ \frac{a_1^2 b_1 \ell_1 \ell_2 x_2 \sin x_3}{a_2^2 - a_1 b_2 \ell_2^2} \end{bmatrix} (P) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(\text{ad}X^0, X^2)(P) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Thus, $\dim R^2 = 4$

and $\lambda_2 = \dim R^2 - \dim R^1 = 2$

$$\lambda_i = 0 \quad \text{for } i \geq 3$$

The Kronecker indices are:

$$K_1 = \{\# \lambda_i \mid \lambda_i \geq 1\} = 2$$

$$K_2 = \{\# \lambda_i \mid \lambda_i \geq 2\} = 2$$

$$K_i = 0 \quad i \geq 3$$

all less than or equal to 2.

For the conditions of the feedback linearization theorem, since $X^0(P) = 0$, the first condition is trivial. We showed that all Kronecker indices are less than or equal to two. This makes condition 3 always satisfied. For condition 2, let

$$C = \{X^1, (\text{ad}X^0, X^1), X^2, (\text{ad}X^0, X^2)\}$$

then $\dim C(P) = \dim R^2 = 4$. Hence condition 2 is true. Therefore system (4.38) is feedback equivalent to the canonical form (4.40)

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = u_1 \\ \dot{y}_3 = y_4 \\ \dot{y}_4 = u_2 \end{cases} \quad (4.40)$$

By inspection of the vector fields X^1, X^2 ; the real functions h_1 and h_2 are chosen to be

$$h_1(x) = x_1 \quad (4.41)$$

$$h_2(x) = x_3$$

such that,

$$h_1(0) = h_2(0) = 0$$

$dh_1(0), dh_2(0)$ linearly independent

$$L_{X^i} h_j = 0 \quad i, j = 1, 2$$

The coordinate transform is then,

$$y_1 = h_1(x) = x_1$$

$$y_2 = (L_{X^0} h_1)(x) = \langle X^0, (1, 0, 0, 0)^T \rangle = x_2$$

$$y_3 = x_3$$

$$y_4 = (L_{X^0} h_2)(x) = \langle X^0, (0, 0, 1, 0)^T \rangle = x_4$$

which is the identity.

The feedback controls $u_1(x)$, $u_2(x)$ as given by (4.26) are:

$$u_1(x) = -f_1(x) + K_1(w_1 - x_1) - K_2x_2 \quad (4.42)$$

$$u_2(x) = -f_2(x) + K_3(w_3 - x_3) - K_4x_4 \quad (4.43)$$

where K_1, K_2, K_3, K_4 are feedback gains and w_1, w_3 are the desired joint positions.

The actual control torques to be applied by the actuators are obtained by solving equations (4.34) and (4.37) for T_1 and T_2 . We finally have:

$$\begin{aligned} T_1 = & a_1(x) \left(-f_1(x) + K_1(w_1 - x_1) - K_2x_2 - g_1(x) \right) \\ & + a_2(x) \left(-f_2(x) + K_3(w_3 - x_3) - K_4x_4 - g_2(x) \right) \end{aligned} \quad (4.44)$$

$$\begin{aligned} T_2 = & a_2(x) \left(-f_1(x) + K_1(w_1 - x_1) - K_2x_2 - g_1(x) \right) \\ & + b_2 \ell_2^2 \left(-f_2(x) + K_3(w_3 - x_3) - K_4x_4 - g_2(x) \right) \end{aligned} \quad (4.45)$$

When these torques T_1 and T_2 are applied to the joints, the nonlinear system (4.31) (robot model) has the same dynamic behavior as the uncoupled second order linear subsystems (4.40).

For this two joint manipulator, a computer simulation was performed to evaluate the performance of the controller.

We simulated the hand motion from one path point to the next point, or through the inverse kinematics, from one state of joint positions to a desired state. Initial data ($\theta_1 = 1$ rd, $\theta_2 = -0.5$ rd) and desired joint positions ($\theta_1 = 0.5$ rd, $\theta_2 = 0$ rd) were arbitrarily chosen.

Table 4.6. Simulation of joint motions from (1,-0.5) to (0.5,0)
with $K_1 = K_3 = 400$.

TWO JOINT ROBOT MANIPULATOR-FEEDBACK LINEARIZATION
CONSTANT FEEDBACK GAINS

TIME	THETA1	THETA2
0	1.00000	-.50000
.10000	.70479	-.20479
.20000	.54773	-.04773
.30000	.50957	-.00957
.40000	.50180	-.00180
.50000	.50033	-.00033
.60000	.50006	-.00006
.70000	.50001	-.00001
.80000	.50000	-.00000

TWO JOINT MANIPULATOR

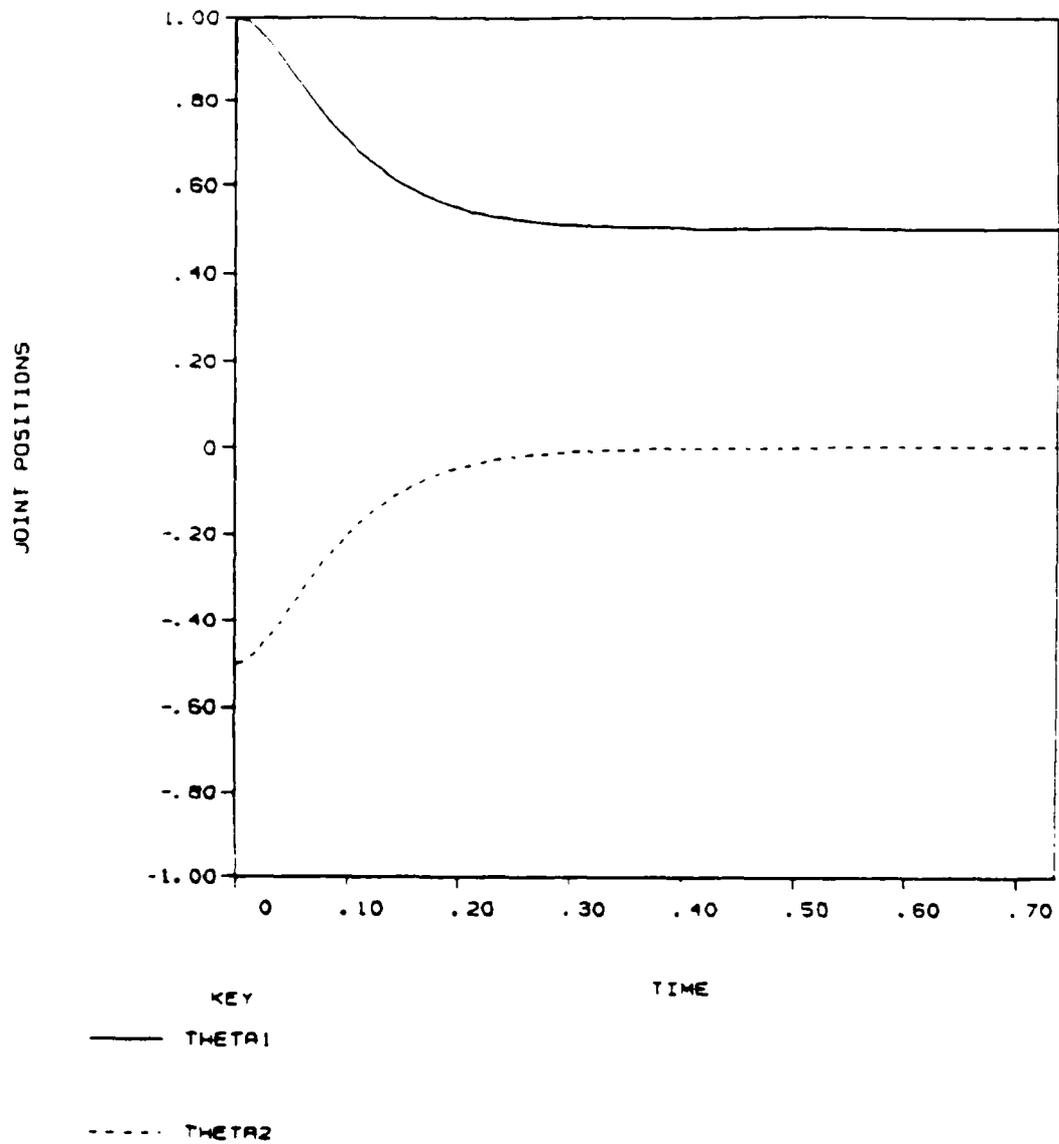


Figure 4.6. Feedback linearization. Simulation of joint motions from (1,-0.5) to (0.5,0).

TWO JOINT MANIPULATOR

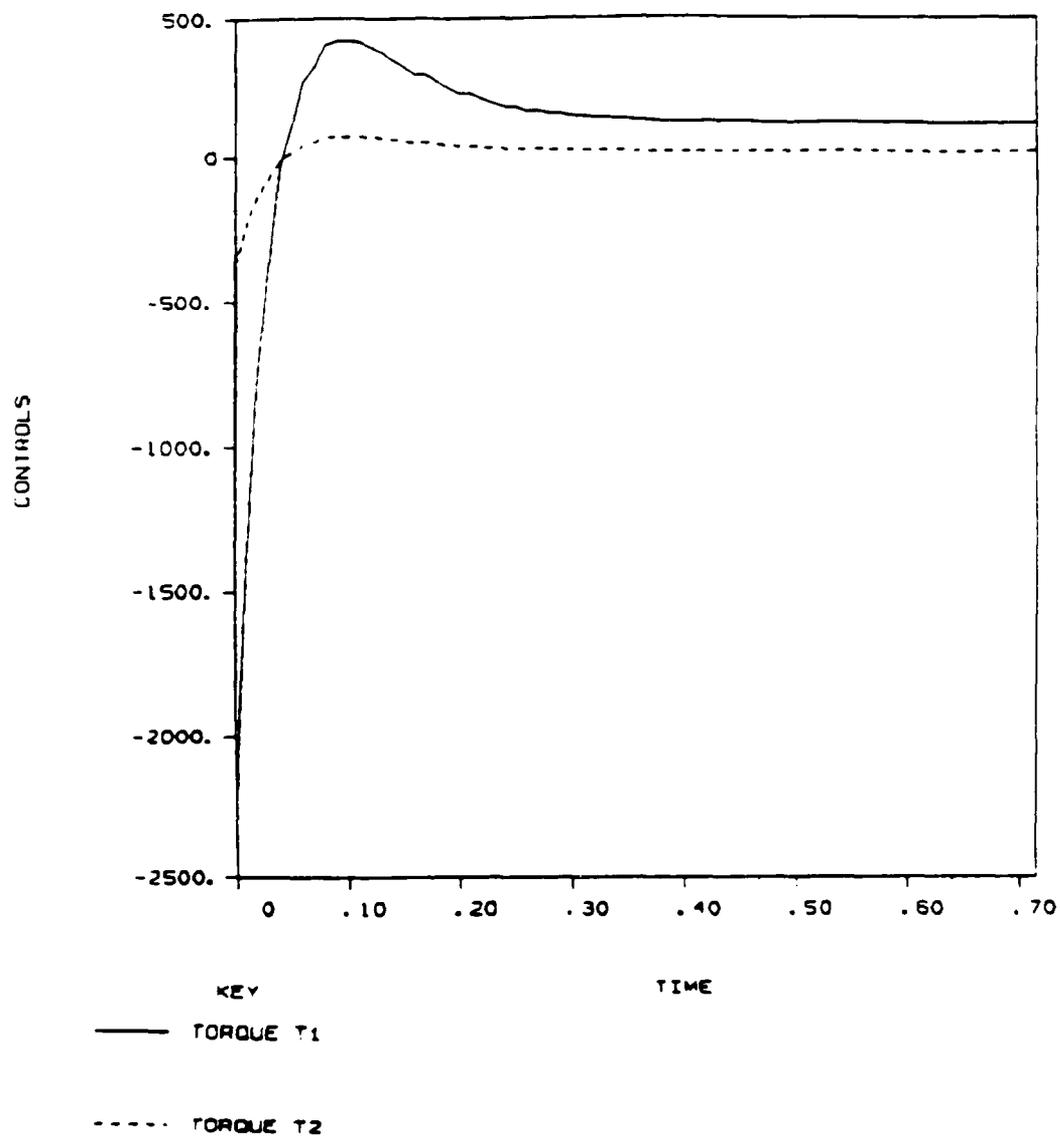


Figure 4.7. Feedback linearization. Controls.

The feedback gains K_1 , K_2 , K_3 and K_4 were chosen for a critically damped response. One such a choice is: $K_1 = K_3 = 400$, $K_2 = K_4 = 40$. The simulation results are presented in figures 4.6, 4.7 and table 4.6. These results show a good performance. There are no steady state errors and no overshoot. The convergence time is 0.75 s and the maximum torque is -2212 Nm.

A more thorough computer simulation analysis will be presented for the three joint manipulator in order to be able to compare with the computed torque technique.

Example 4.2.3: Three joint manipulator

The dynamic equations of the three joint manipulator (3.11)-(3.13) can be rewritten as:

$$\begin{aligned}\ddot{r} &= \frac{b_2}{b_1} r \dot{\phi}^2 - \frac{3}{8} \frac{b_4}{b_1} \dot{\phi}^2 - \frac{b_3}{b_1} g \sin \phi + \frac{F_R}{b_1} \\ \ddot{\phi} &= -2b_2 \frac{r \dot{\phi}}{c_1} + \frac{3}{4} b_4 \frac{\dot{\phi}}{c_1} - b_3 g \frac{r \cos \phi}{c_1} + \frac{1}{2} b_4 g \frac{\cos \phi}{c_1} + \frac{T}{c_1} \\ \ddot{z} &= \frac{F_z}{b_3}\end{aligned}\tag{4.46}$$

where

$$\begin{aligned}b_1 &= m_L + \frac{m_R}{4} \\ b_2 &= m_L + \frac{3}{4} m_R \\ b_3 &= m_L + m_R \\ b_4 &= m_R^2\end{aligned}\tag{4.47}$$

and

$$c_1 = b_2 r^2 + \frac{1}{4} b_4 (\ell - 3r) . \quad (4.48)$$

Let the state variables be:

$$\begin{aligned} x_1 &= r & x_3 &= \vartheta & x_5 &= z \\ x_2 &= \dot{r} & x_4 &= \dot{\vartheta} & x_6 &= \dot{z} \end{aligned} \quad (4.49)$$

The state representation of (4.46) is then,

$$\left\{ \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(x) + u_1(x) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(x) + u_2(x) \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= u_3(x) \end{aligned} \right. \quad (4.50)$$

where

$$c_1(x) = b_2 x_1^2 + \frac{1}{4} b_4 (\ell - 3x_1) \quad (4.51)$$

$$f_1(x) = \frac{b_2}{b_1} x_1 x_2^2 - \frac{3}{8} \frac{b_4}{b_1} x_4^2 \quad (4.52)$$

$$u_1(x) = -\frac{b_3}{b_1} g \sin x_3 + \frac{F_R}{b_1} \quad (4.53)$$

$$f_2(x) = -2b_2 \frac{x_1 x_2 x_4}{c_1(x)} + \frac{3}{4} b_4 \frac{x_2 x_4}{c_1(x)} \quad (4.54)$$

$$u_2(x) = -b_3 g \frac{x_1 \cos x_3}{c_1(x)} + \frac{1}{2} b_4 g \frac{\cos x_3}{c_1(x)} + \frac{\Gamma}{c_1(x)} \quad (4.55)$$

$$u_3(x) = \frac{Fz}{b_3} \quad (4.56)$$

System (4.50) can be expressed as:

$$\dot{x} = X^0(x) + X^1(x)u_1(x) + X^2(x)u_2(x) + X^3(x)u_3(x) \quad (4.57)$$

where

$$X^0 = \begin{bmatrix} x_2 \\ f_1(x) \\ x_4 \\ f_2(x) \\ x_6 \\ 0 \end{bmatrix}, \quad X^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad X^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad X^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$u_1(x)$, $u_2(x)$, $u_3(x)$ are the controls and $P = 0$ is the equilibrium point of (4.50).

We check again the Kronecker indices.

$$l_1 = \dim R^1 = \dim \{X^1(P), X^2(P), X^3(P)\} = 3$$

let

$$R^2 = \{X^1(P), (\text{ad}X^0, X^1)(P), X^2(P), (\text{ad}X^0, X^2)(P), X^3(P), (\text{ad}X^0, X^3)(P)\}$$

$$(\text{ad}X^0, X^1)(P) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -2b_2 \frac{x_1 x_4}{c_1} + \frac{3}{4} b_4 \frac{x_4}{c_1} \\ 0 \\ 0 \end{bmatrix} (P) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(\text{ad}X^0, X^2)(P) = \begin{bmatrix} 0 \\ 2\left(\frac{b_2}{b_1} x_1 - \frac{3}{8} \frac{b_4}{b_1}\right) x_4^2 \\ 1 \\ -2b_2 \frac{x_1 x_2}{c_1} + \frac{3}{4} b_4 \frac{x_2}{c_1} \\ 0 \\ 0 \end{bmatrix} (P) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(\text{ad}X^0, X^3)(P) = (0, 0, 0, 0, 1, 0)^T$$

Hence $\dim R^2 = 6$ and $\ell_2 = \dim R^2 - \dim R^1 = 3$, $\ell_i = 0$ $i \geq 3$.

The Kronecker indices are given by:

$$K_1 = K_2 = K_3 = 2, \quad K_i = 0 \quad i \geq 4$$

Thus conditions 1 and 3 of the feedback theorem are satisfied.

Also $C(P) = R^2$. Thus, $\dim C(P) = 6$, which makes condition 3 true.

System (4.50) is then feedback equivalent to the following linear canonical form (4.58):

$$\left\{ \begin{array}{l} \dot{y}_1 = y_2 \\ \dot{y}_2 = \mu_1 \\ \dot{y}_3 = y_4 \\ \dot{y}_4 = \mu_2 \\ \dot{y}_5 = y_6 \\ \dot{y}_6 = \mu_3 \end{array} \right. \quad (4.58)$$

where μ_1 , μ_2 and μ_3 are new free controls. Again, the real valued functions h_1 , h_2 , h_3 are chosen as:

$$\begin{aligned} h_1(x) &= x_1 \\ h_2(x) &= x_3 \\ h_3(x) &= x_5 \end{aligned} \quad (4.59)$$

which leads to an identity coordinate change. The feedback controls u_1 , u_2 and u_3 given by (4.26) are:

$$\begin{aligned} u_1(x) &= -f_1(x) + K_1(w_1 - x_1) - K_2 x_2 \\ u_2(x) &= -f_2(x) + K_3(w_3 - x_3) - K_4 x_4 \\ u_3(x) &= K_5(w_5 - x_5) - K_6 x_6 \end{aligned}$$

The explicit control forces F_R , F_z and torque T are obtained from (4.53), (4.55) and (4.56).

$$F_R = b_1[-f_1(x) + K_1(w_1 - x_1) - K_2 x_2] + b_3 g \sin x_3 \quad (4.63)$$

$$T = c_1(x)[-f_2(x) + K_3(w_3 - x_3) - K_4 x_4] + b_3 g x_1 \cos x_3 - \frac{1}{2} b_4 g \cos x_3 \quad (4.64)$$

$$F_z = b_3[K_5(w_5 - x_5) - K_6 x_6] \quad (4.65)$$

These control forces F_R , F_z and control torque T , transform the nonlinear system (4.50) (robot model) into the linear canonical system (4.58).

Computer simulations were conducted in the same conditions as for the 'computed torque' technique. We first simulated the joint motions from initial positions ($r = 1$ m, $\phi = 0$ rd, $z = -1$ m) to desired positions ($r = 0.5$ m, $\phi = -0.6$ rd, $z = -0.2$ m). $K_1 = K_3 = K_5 = 400$ and $K_2 = K_4 = K_6 = 40$, were chosen for a critically damped response. The simulation results are presented in table 4.7 and figures 4.8 and 4.9. It is noted that there are no steady state errors and no overshoot. The largest control effort required is $F_z = 4480$ Newtons. The convergence time (time at which the steady state is reached) is 0.7 seconds. This convergence time can be further decreased by choosing larger feedback gains, as table 4.8 shows. However, this will result in much larger control torques and forces. For the 'computed torque' technique, as was discussed previously, with a feedback gain $K_p = 5000$ we observed a position error of 14 mm (for r and z) and a maximum control force $F_z = 56993$ N. So, clearly there is an appreciable improvement in performance with the feedback linearization approach.

Table 4.7. Simulation of joint motions from (1,0,-1) to
(0.5,-0.6,-0.2).

THREE JOINT ROBOT MANIPULATOR-FEEDBACK LINEARIZATION
CONSTANT FEEDBACK GAINS.

TIME	R	PHI	Z
0	1.00000	0	-1.00000
.10000	.70480	- .35424	- .52767
.20000	.54773	- .54272	- .27638
.30000	.50857	- .58851	- .21532
.40000	.50180	- .59794	- .20288
.50000	.50033	- .59960	- .20053
.60000	.50006	- .59993	- .20010
.70000	.50001	- .59999	- .20002
.80000	.50000	- .60000	- .20000

Table 4.8. Position error, maximum force F_z and convergence time
as a function of gain K_1 .

Feedback gain, K_1	Position error	$(F_z)_{\max}$ (N)	Convergence time (s)
200	0	2240	0.85
400	0	4480	0.6
800	0	8960	0.52
1500	0	16800	0.36
3000	0	33600	0.29

THREE JOINT MANIPULATOR

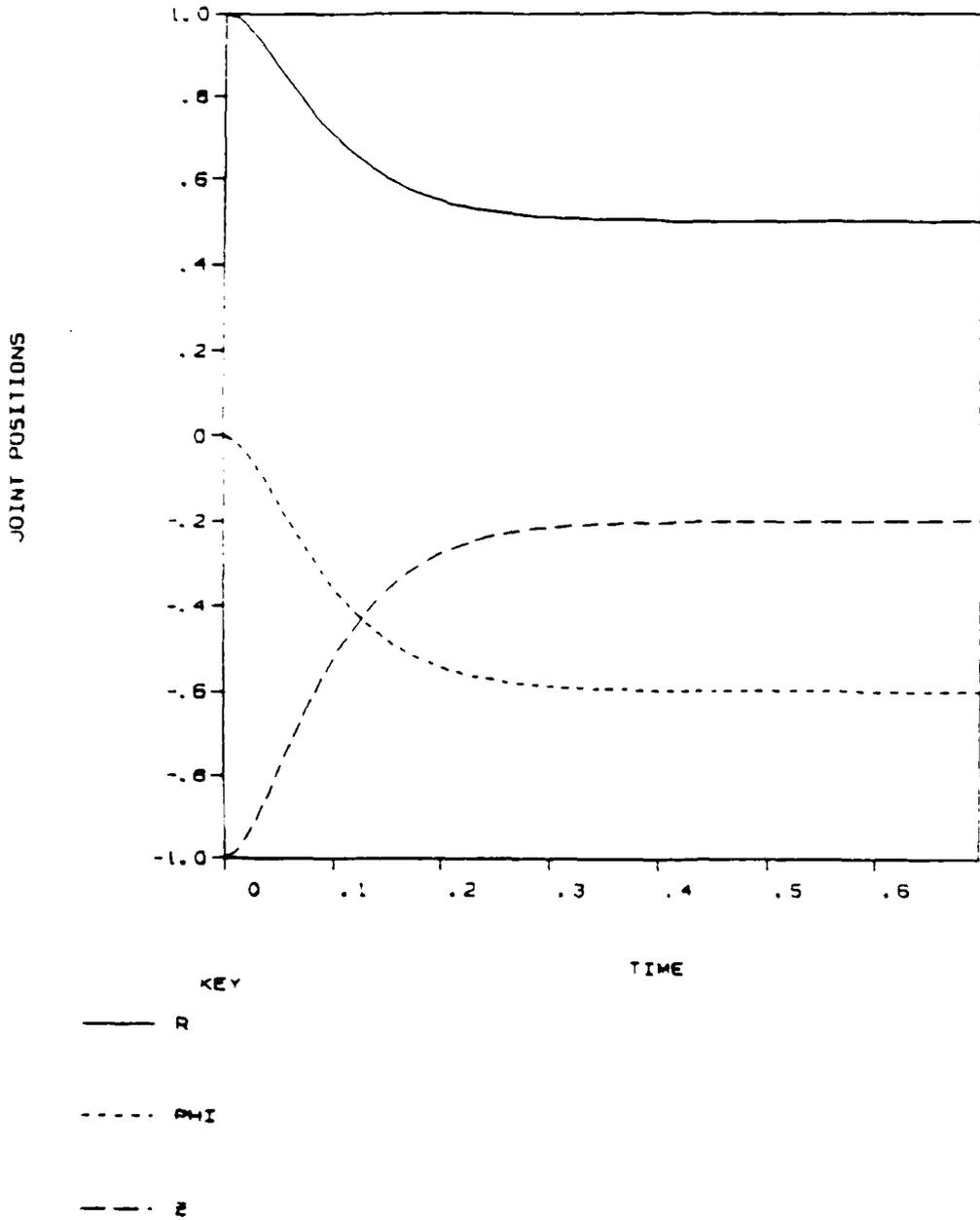


Figure 4.8. Feedback linearization. Simulation of joint motions from (1,0,-1) to (0.5,-0.6,-0.2).

THREE JOINT MANIPULATOR

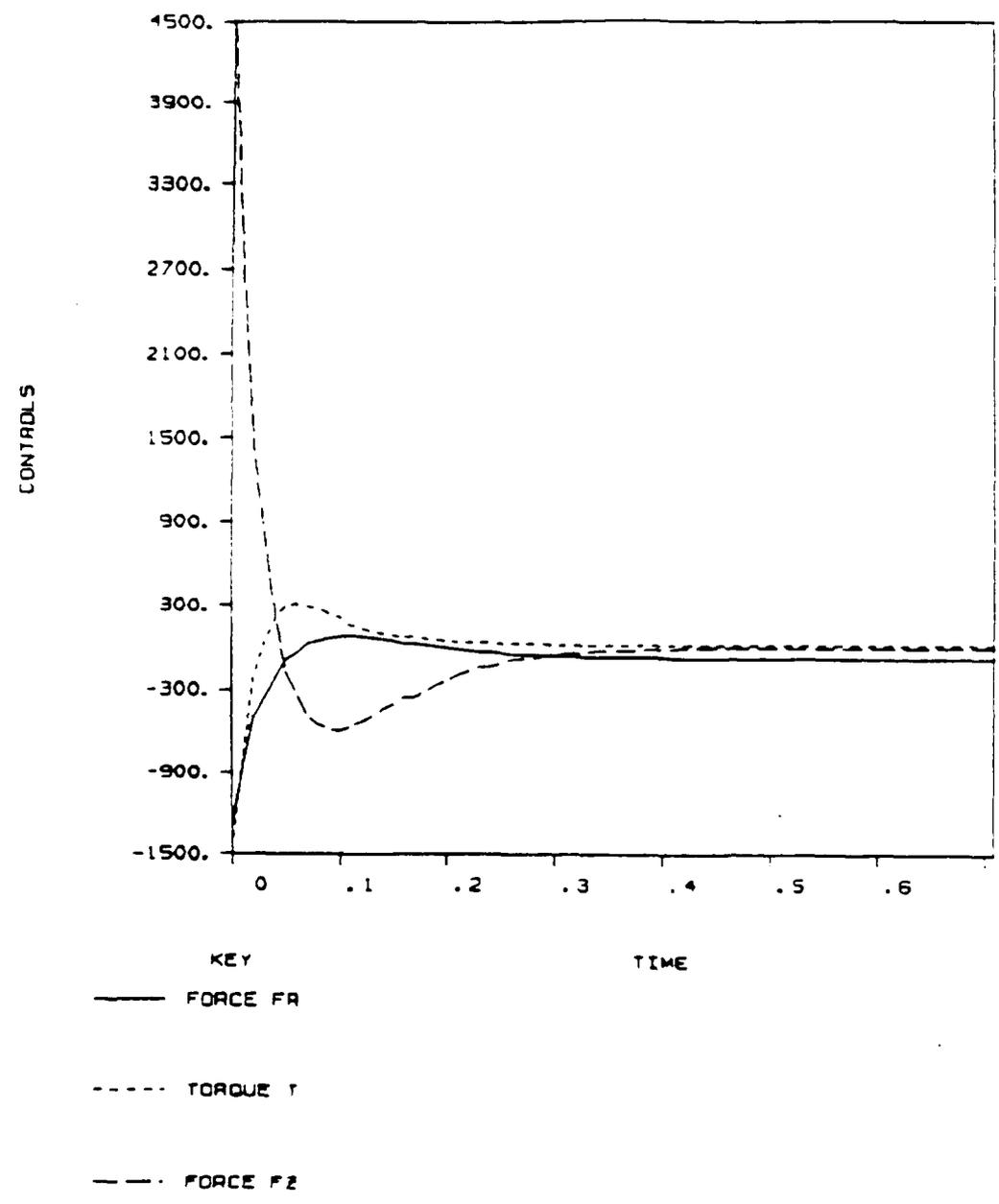


Figure 4.9. Feedback linearization. Controls.

In the computed torque technique, we have demonstrated that it was possible to lower the control effort requirements by utilizing time varying feedback gains. The same idea can be applied to the feedback linearization. We first specify the maximum permissible control torques and forces, and start the joint motions with small feedback gains. Then as the error positions are diminishing, these gains are increased in a piecewise manner [34].

In this study, we applied this technique to the three joint manipulator. The maximum torque (force) is chosen to be 400 N.m (N) and the feedback gains are doubled as long as the torque limits are not exceeded. Similarly, if the limits are about to be exceeded, the feedback gains are reduced by half. Again, we simulated the joint motions from ($r = 1$ m, $\phi = 0$ rd, $z = -1$ m) to ($r = 0.5$ m, $\phi = -0.6$ m, $z = -0.2$ m). The simulation results are given in table 4.9a and figures 4.10-4.11. Note a shorter convergence time of 0.5 seconds, with no overshoot and no steady state errors. This was achieved with only 10% (400 vs. 4480) of the control effort required by the constant feedback gain design.

One may try to optimize this technique by using the maximum torques and forces available, which would result in an even faster response. Such an optimization can be accomplished by feedback gains which vary by a small increment. This increment would depend on the control limits; and the smaller this step size, the faster the response. However, as the simulations showed, if the increment is too small, the system can acquire a large inertia. Since the controls are bounded, this could result in overshoot.

Table 4.9a. Simulation of joint motions from (1,0,-1)
to (0.5,-0.6,-0.2).

THREE JOINT ROBOT MANIPULATOR-FEEDBACK LINEARIZATION.
TIME VARYING FEEDBACK GAINS.

TIME	R	PHI	Z
0	1.00000	0	-1.00000
.05000	.98460	-.01848	-.97536
.10000	.94298	-.06342	-.90877
.15000	.88646	-.13625	-.81833
.20000	.81122	-.22653	-.69795
.25000	.73272	-.32074	-.57235
.30000	.65193	-.41769	-.44308
.35000	.57964	-.50443	-.32743
.40000	.52615	-.56862	-.24184
.45000	.50107	-.59872	-.20171
.50000	.50000	-.60000	-.20000

Table 4.9b. Feedback linearization with optimal time
varying feedback gains. Simulation of
joint motions.

THREE JOINT ROBOT MANIPULATOR-FEEDBACK LINEARIZATION.
TIME VARYING FEEDBACK GAINS.

TIME	R	PHI	Z
0	1.00000	0	-1.00000
.05000	.98194	-.02168	-.97110
.10000	.93023	-.08373	-.88836
.15000	.84914	-.18103	-.75862
.20000	.74474	-.30632	-.59158
.25000	.63051	-.44339	-.40882
.30000	.54812	-.54226	-.27699
.35000	.50648	-.59223	-.21036
.40000	.50014	-.59983	-.20022
.45000	.50000	-.60000	-.20000

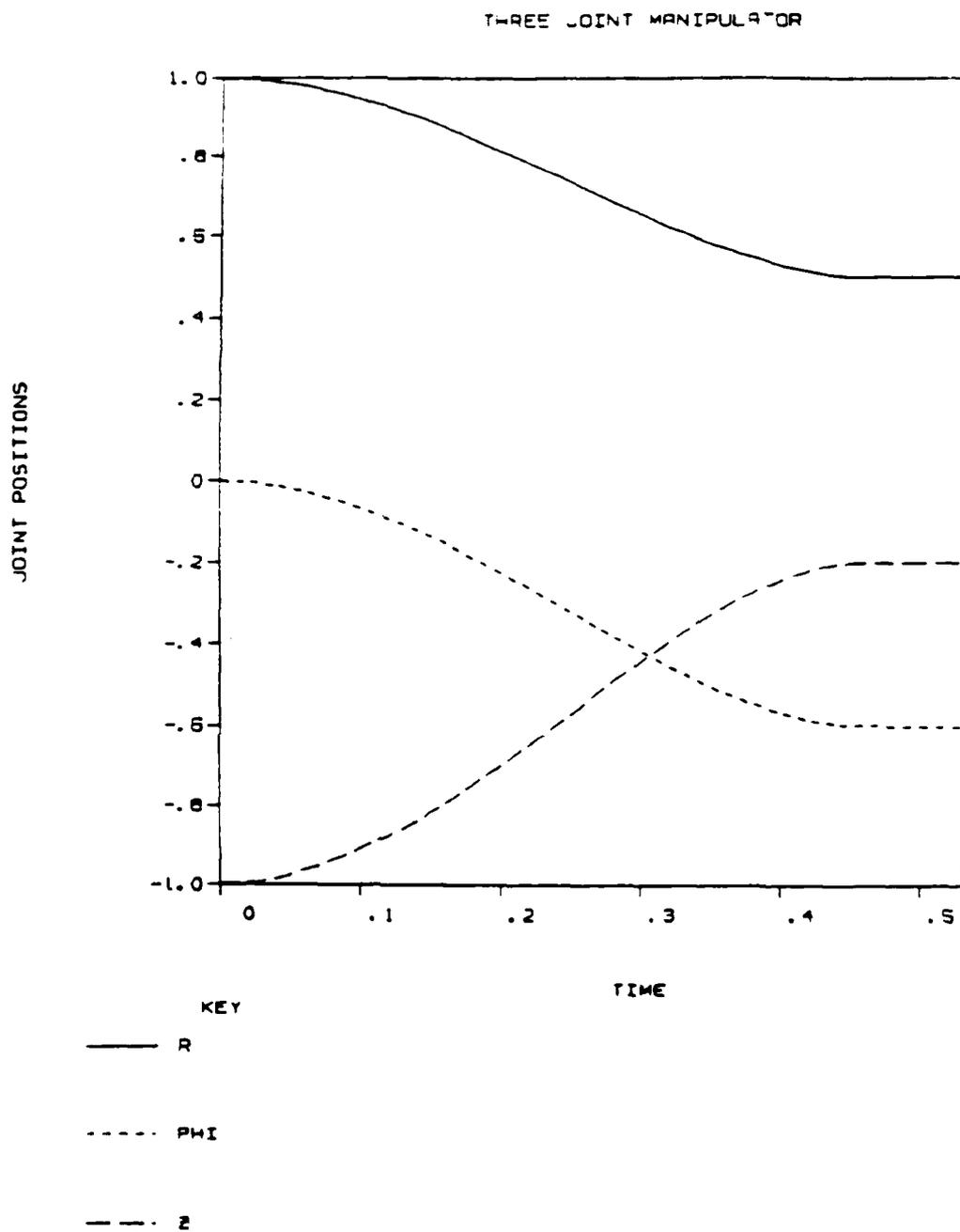


Figure 4.10. Feedback linearization with time varying feedback gains. Simulation of joint motions from $(1,0,-1)$ to $(0.5,-0.6,-0.2)$.

THREE JOINT MANIPULATOR

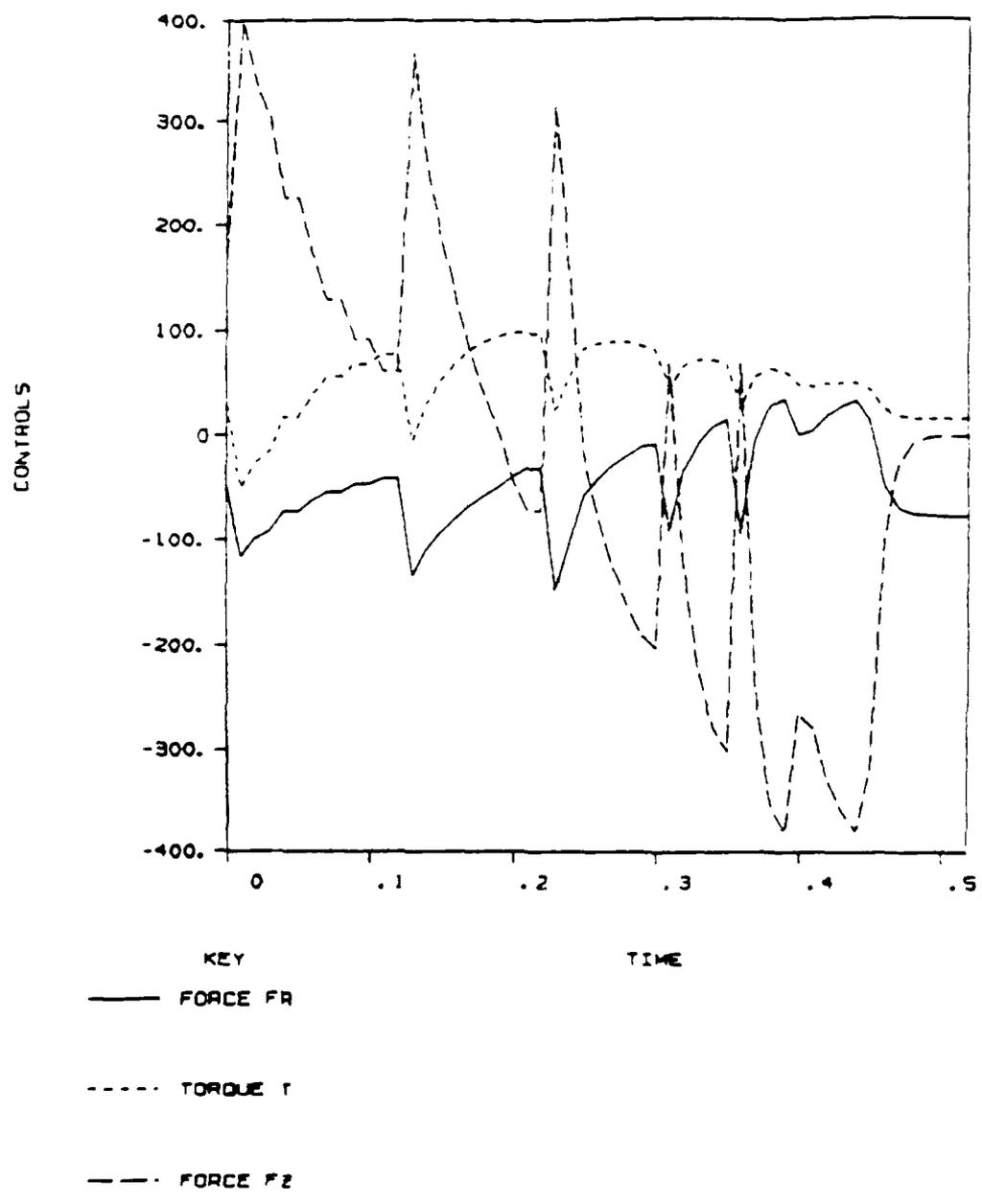


Figure 4.11. Feedback linearization with time varying feedback gains. Controls.

THREE JOINT MANIPULATOR

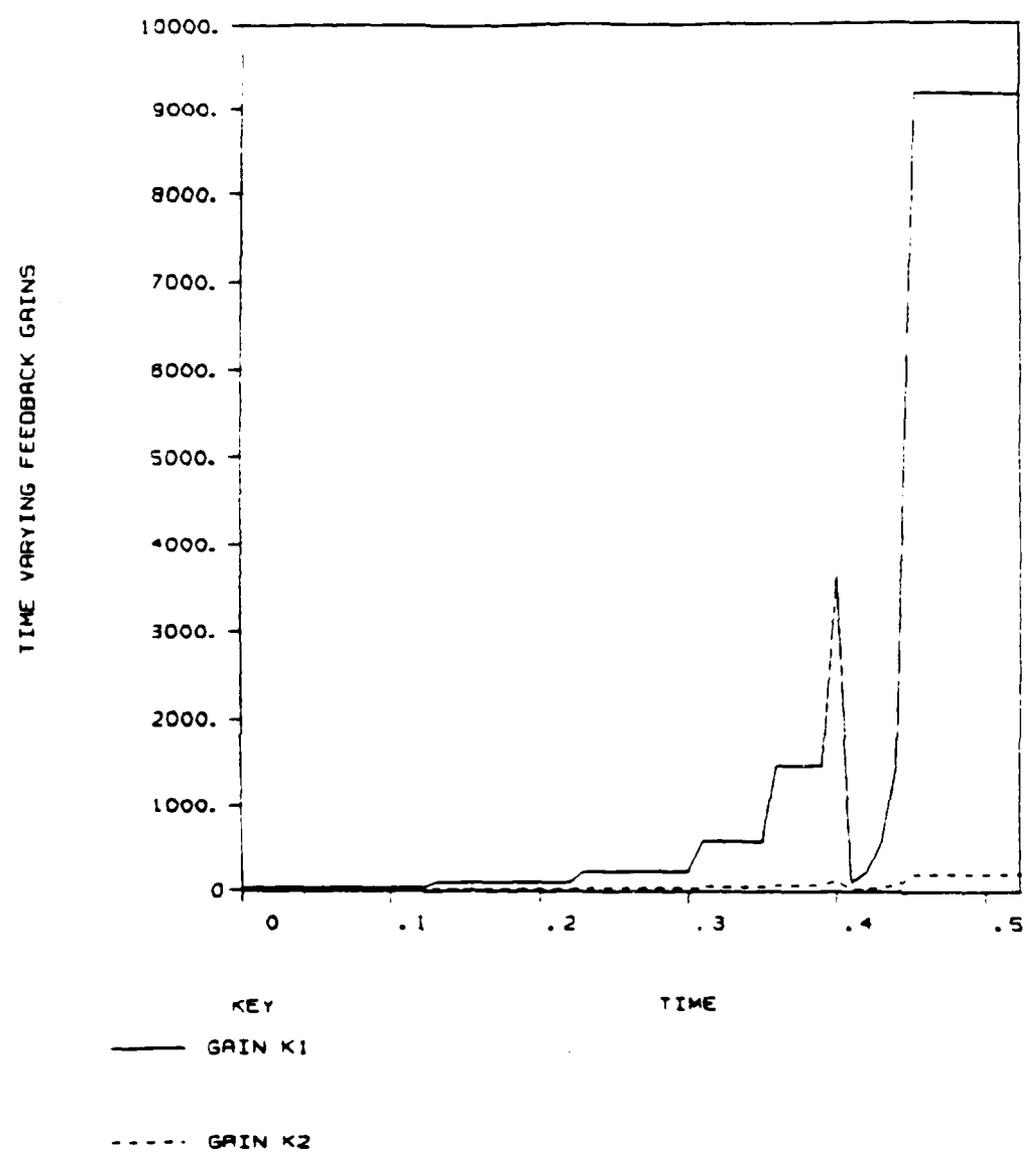


Figure 4.12. Feedback linearization. Time varying feedback gains K_1 , K_2 .

THREE JOINT MANIPULATOR

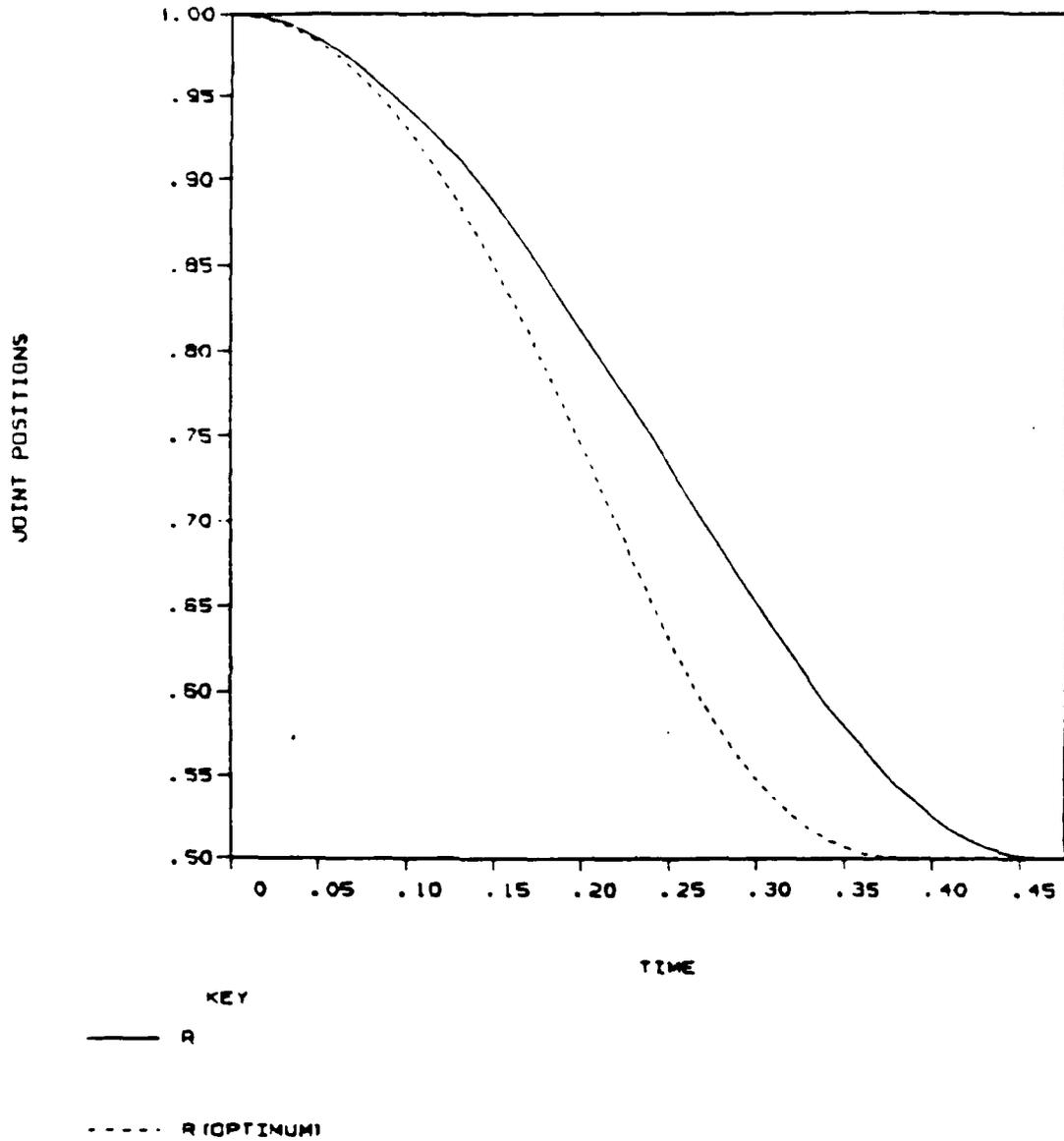


Figure 4.13. Feedback linearization. Optimal time varying feedback gains versus feedback gains which are varied by a step size of one.

THREE JOINT MANIPULATOR

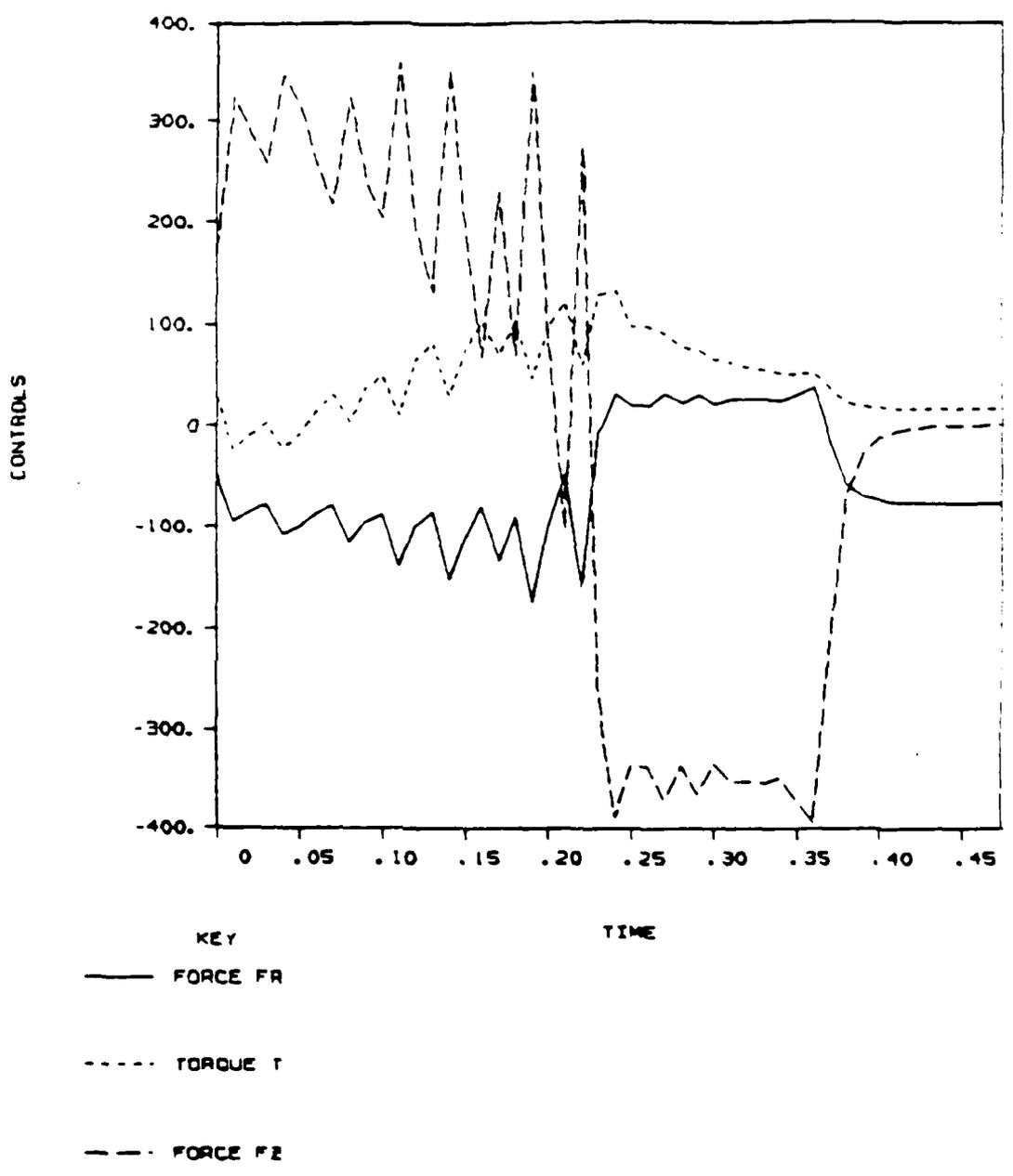


Figure 4.14. Feedback linearization with optimal time varying feedback gains. Controls.

Using this optimal approach, we simulated the same joint motions with a torque (force) limit of 400 N.m (N). For these control bounds, a step size of 0.5 was used. To avoid the overshoot problem, we switched from a critically damped response to an overdamped response in the vicinity of the final desired positions. The simulation results given in table 4.9b and figures 4.13-4.14 show a decrease in convergence time of approximately 0.1 second (0.4 vs. 0.5) for this optimal approach over the case when the feedback gains were doubled.

This optimal approach needs further investigation in order to determine the exact relationship between the feedback gain increment and the control bounds, with the overshoot as a constraint.

A control system is said to be robust if it can accommodate disturbances, parameter variations, and model inaccuracies. We tested the robustness of our nonlinear feedback controller by introducing an error in the actual computed control torques. As table 4.10 illustrates, up to 60% error still gives a good trajectory tracking. So, the controller is quite robust.

Finally, we simulated the replanned path motion of table 4.4 for the nonlinear feedback controller and the 'computed torque' controller. The time varying feedback gain technique was used in both cases. For the nonlinear controller, we specified a maximum allowable torque (force) of 400 Nm (N). For the 'computed torque' technique we permitted a maximum torque (force) of 5000 Nm (N).

Table 4.10. Effect of disturbances and parameter variations on the tracking accuracy.

Percent of error in controls	Final position error		
	Δr (mm)	$\Delta \phi$ (mrd)	Δz (mm)
10	0.08	0.003	0
20	0.15	0.006	0
30	0.3	0.02	0.001
60	1.2	0.2	0.1

Table 4.11. Path tracking simulation.

Joint	Feedback linearization $T_{\max} = 400 \text{ Nm}$		'Computed torque' $T_{\max} = 5000 \text{ Nm}$	
	Maximum tracking error	Final position error	Maximum tracking error	Final position error
Δr (mm)	2	0	11	4
$\Delta \phi$ (mrd)	2	0	10	4
Δz (mm)	3	0	11	4

Comparative results are presented in table 4.11. We note that the feedback linearization design yields a smaller tracking error and no final position error. However, the 'computed torque' design results in larger tracking error and a steady state final position error. Again, it appears that the feedback linearization approach has a better overall performance. This approach combined with the time varying design results in fast and accurate tracking with relatively low energy control requirements.

CHAPTER V

CONCLUSIONS

The robot control problem in general consists of planning trajectories which describe desired hand motions, and then controlling the robot to ensure that those trajectories are correctly executed. In order to move the hand along a trajectory, control torques must be exerted by actuators at the joints. Thus, it is necessary to convert the desired hand trajectories into a time sequence of desired joint coordinates (inverse kinematics).

The dynamics of a n -joint manipulator is very complex. The resulting dynamic mathematical model is a system of n highly nonlinear and coupled second order differential equations. For a six-joint manipulator, this system can contain hundreds of algebraic terms. The dynamic control of such a system is then the problem of controlling a multi-input nonlinear system. For robots, many control strategies have been developed, among which the commonly used open loop control is the simplest.

In this thesis, we presented and analyzed a widely used independent joint control method, the 'computed torque' technique. This approach is basically a PD action law with some nonlinear compensations. The simulation study confirmed the fact that the independent joint control requires high-power actuators for

adequate path tracking. We demonstrated that it is possible to lower this requirement by using time varying feedback gains.

In this thesis, we also have studied a nonlinear control approach based on feedback linearization. The feedback linearization is a global linearization of a nonlinear system via a coordinate change and feedbacks. A recent development in this theory is a theorem which gives necessary and sufficient conditions to linearize a nonlinear system [25].

To design the nonlinear controller, we first began with the derivation of the state space representation of a n -joint manipulator dynamic equations. This state representation led to Kronecker indices all less than or equal to 2. We then showed that the three necessary and sufficient conditions of the feedback linearization theorem are all satisfied. The fact that all Kronecker indices are all less than or equal to 2 resulted in a linear canonical form composed of n linear, time invariant, second order, uncoupled subsystems. The control problem was then reduced from controlling a multi-input nonlinear system to controlling n uncoupled linear second order subsystems. Each subsystem was then stabilized by state feedback. Finally, we constructed a general feedback control algorithm, which can be implemented on a computer.

We illustrated this approach by two design examples. Computer simulations were also conducted to analyze and evaluate the performance of the nonlinear controller. The simulation results show satisfactory performances. We obtained a fast

response with no steady state error and no overshoot. The performance was further improved by utilizing the time varying feedback gain technique. The maximum required control torques and forces were reduced and the response was faster.

In practice, it is desirable that a control system can reject disturbances, parameter variations and model inaccuracies. We tested the robustness of the controller by introducing an error in the actual computed control torques and forces. It was found that the controller is indeed robust. This may suggest that if on-line computation of the controls is a complex problem, the dynamic equations (used in control computations) can be simplified without loss in dynamic performance.

Also, a path tracking motion in joint space was simulated for both the feedback linearization design and the 'computed torque' technique design. The simulation results show a substantially better performance for the feedback linearization approach, which yields smaller tracking error, no final position error and lower control effort requirements.

Of course, it would be interesting to see the results of a practical implementation of this nonlinear control scheme. However, this implementation is not done in this thesis due to lack of robot hardware. It will be done when the robot hardware is available to us in the future.

We also mention that in spite of the complexity of the dynamic equations of a n -joint robot manipulator the

AD-A188 094

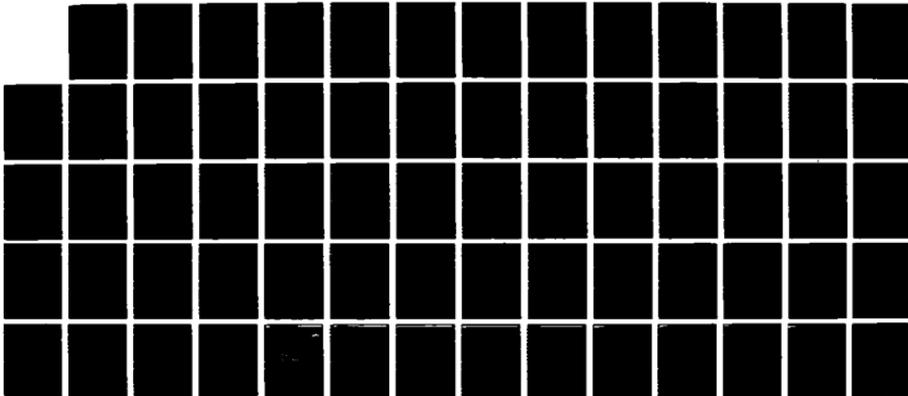
TIME VARYING CONTROLLER DESIGN FOR ROBOT MANIPULATOR
CONTROL(U) COLORADO UNIV AT BOULDER M Y HU ET AL.
25 AUG 87 CU-1536900 N00014-84-K-8425

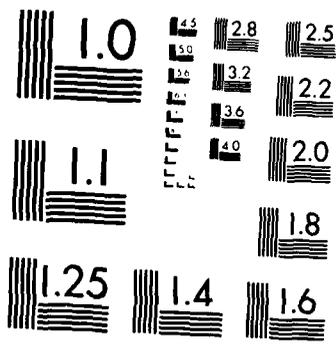
2/2

UNCLASSIFIED

F/G 13/9

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

linearization theorem was readily applicable. This is mainly due to the fact that all Kronecker indices are all less than or equal to two. Because, in general, the conditions for linearization are restrictive. First of all the involutive condition. In terms of nonlinear system theory in general, one may try to extend the applicability of the feedback linearization theorem by seeking other canonical forms, in addition to the linear canonical form (4-2-3). For example, a nilpotent Lie algebra with a special structure. This can be a future research problem.

REFERENCES

- [1] D.E. Whitney, 'Resolved motion rate control of manipulators and human prostheses.' IEEE Transactions MMS-10, 1969.
- [2] M.E. Kahn and B. Roth, 'The near-minimum-time control of open loop articulated kinematics chains.' Trans. ASME, Journal of Systems, Measurement and Control 93, 1971.
- [3] J.F. Engleberger, Robotics in practice: Management and applications of industrial robots. Amacon, 1980.
- [4] J.Y.S. Luh, W.D. Fisher, R.P. Paul, 'Joint torque control by a direct feedback for industrial robots.' Proc. 20th IEEE Conf. on Decision and Control, San Diego, CA, December 1981.
- [5] D.F. Golla, S.c. Garg, and P.C. Hughes, 'Linear state-feedback control of manipulators.' Mech. Machine Theory 16, 1981.
- [6] M.H. Raibert, J.J. Craig, 'Hybrid position/force control of manipulators.' J. Dynamic Systems, Measurement and Control 102, June 1981.
- [7] S. Dubowsky, D.T. DesForges, 'The applications of model-referenced adaptive control to robotic manipulators.' J. Dynamic Systems, Measurement and Control 101, 1979.
- [8] D.E. Whitney and E.F. Junkel, 'Applying stochastic control theory to robot sensing, teaching, and long term control.' Proc. Amer. Control Conf., Alexandria, VA, June 1982.
- [9] M. Takegaki, S. Arimoto, 'A new feedback method for dynamic control of manipulators.' J. Dynamic Systems, Measurement and Control 102, 1981.
- [10] C.S.G. Lee, M.J. Chung, 'An adaptive control strategy for mechanical manipulators.' IEEE Trans. on Autom. Control AC-29, September 1984.
- [11] T. Eun, Y.J. Cho, H.S. Cho, 'Stability and positioning accuracy of a pneumatic on-off servo mechanism.' Proc. Amer. Control Conf., Alexandria, VA, June 1982.

- [12] T.L. Johnson, 'On feedback laws for robotic systems.' Proc. 8th IFAC World Congress, Japan, August 1981.
- [13] K.K.D. Young, 'Controller design for a manipulator using theory of variable structure systems.' IEEE Trans. Systems, Man, Cybernetics SMC-8, 1978.
- [14] A. Isidori, A.J. Krener, C. Gori-Giorgi, and S. Monaco, 'Nonlinear decoupling via feedback: A differential geometric approach.' IEEE Trans. Autom. Control AC-26, 1981.
- [15] W. Respondek, 'On decomposition of nonlinear control systems.' Systems and Control Letters 1, 1982.
- [16] A.J. Krener, 'On the equivalence of control systems and linearization of nonlinear systems.' SIAM J. Control and Opt. 11, 1973.
- [17] J.L. Sedwick, D.L. Elliott, 'Linearization of analytic vector fields in the transitive case.' J. of Diff. Eqs. 25, 1977.
- [18] R. Su, 'On the linear equivalence of nonlinear systems.' Systems and Control Letters 2, 1981.
- [19] H. Nijmeijer, A. van der Schaft, 'Controlled invariance for nonlinear systems.' IEEE Trans. AC-27, 1982.
- [20] A. Isidori, 'Lecture notes on differential geometric control theory.' (to appear).
- [21] H. Hemami, P.C. Camana, 'Nonlinear feedback in simple locomotion system.' IEEE Trans. on Autom. Control AC-19, 1976.
- [22] E. Freund, 'Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators.' The International Journal of Robotics Research 1, 1982.
- [23] R.W. Brockett, 'Feedback invariants for nonlinear systems.' IFAC Congress, Helsinki, 1978.
- [24] B. Jakubczyk, W. Respondek, 'On linearization of control systems.' Bull. Acad. Pol. Sci., Ser. Sci. Math. Astronom. Phys. 28, 1980.
- [25] L.R. Hunt, R. Su, G. Meyer, 'Design for Multi-input nonlinear systems.' Differential Geometric Control Theory, Proc. of conf. held at MTU, June-July 1982.
- [26] R.P. Paul, Robot manipulators: Mathematics, Programming, and Control. MIT Press, 1981.

- [27] W.E. Snyder, Industrial robots: Computer interfacing and control. Prentice Hall, 1985 (to appear).
- [28] R.P. Paul, 'Manipulator cartesian path control.' IEEE Trans. on Systems, Man and Cybernetics SMC-9, November 1979.
- [29] L.E. Malvern, Introduction to the mechanics continuous medium. Prentice Hall, 1969.
- [30] B.R. Markiewicz, Analysis of the computed torque drive and comparison with conventional position servo for a computer-controlled manipulator. JPL, Tech. Memo, March 1973.
- [31] A.K. Bejczy, Robot arm dynamics and control. JPL, Tech. Memo, February 1974.
- [32] R.P. Paul, 'Modeling trajectory calculation and servoing of a computer controlled arm.' Stanford ARTif. Intel. Lab., A.I. Memo 177, September 1972.
- [33] M.H. Raibert, B.K.P. Horn, 'Manipulator control using the configuraion space method.' Industrial Robot 5, 1978.
- [34] H. Song, 'The design of linear time-varying system to achieve fast response with no overshoot.' Master's thesis. University of Colorado, Boulder, 1984.
- [35] M. Vukobratovic, 'Dynamics of active articulated mechanisms and synthesis of artificial motion.' Mechanism and Machine Theory 13, 1978.
- [36] Z.X. Wan, Lie Algebras. Pergamon Press, 1975.
- [37] H.J. Sussmann, 'Lie brackets and local controllability: A sufficient condition for scalar-input systems.' SIAM J. Control (to appear).
- [38] H. Hermes, Lecture notes Math 542. University of Colorado, Boulder.
- [39] H. Hermes, 'Local controllability and sufficient conditions in singular problems.' SIAM J. Control and Opt. 14, 1976.
- [40] M. Brady, J.M. Hollerbach, T.L. Johnson, T.L. Perez, M.T. Mason, Robot motion: planning and control. MIT Press, 1982.
- [41] M.Y. Wu, 'Solvability and representation of linear time-varying systems.' Int. J. Control, v. 31, no. 5, 1980.
- [42] T. Kailath, Linear systems. Prentice-Hall, 1980.

APPENDICES

APPENDIX A

```

PROGRAM CRB(INPUT,OUTPUT,TAPE20)
C*****
C      SIMULATION OF A THREE JOINT ROBOT MANIPULATOR-
C      FEEDBACK LINEARIZATION(HUNT-SU-MEYER)-
C      TIME VARYING FEEDBACK GAINS.
C*****

```

```

INTEGER N,METH,MITER,INDEX,IWK(6),IER,K
REAL    Y(6),WK(150),X,TOL,XEND,H,MR,ML,L
COMMON/A/B1,B2,B3,B4,AKR,AM1,AKZ
COMMON/B/Z1,Z2
DIMENSION AZ(120,10),UU(120,10),T(120),Z(120,10)
EXTERNAL FCN,FCNJ
N=6
M=3
X=0.0
Y(1)=1.
Y(2)=0.
Y(3)=0.
Y(4)=0.
Y(5)=-1.
Y(6)=0.
TOL=.00001
H=.0000001
METH=1
MITER=0
INDEX=1
MR=10.
ML=4.
L=1.
Z1=15.
Z2=2.02*SQRT(Z1)
B1=ML+MR/4.
B2=ML+(3./4.)*MR
B3=ML+MR
B4=MR=L
DO 1 I=1,N
    AZ(I,1)=Y(I)
1  CONTINUE
T(1)=0.
Z(1,1)=Z1
Z(1,2)=Z2
DO 30 K=1,60
    XEND=.01*FLOAT(K)
    CALL DGEAR(N,FCN,FCNJ,X,H,Y,XEND,TOL,METH,
    MITER,INDEX,IWK,WK,IER)
    IF(IER.GT.128)GO TO 500
DO 50 I=1,N
    AZ(K+1,I)=Y(I)
50 CONTINUE
UU(K,1)=AKR
UU(K,2)=AM1

```

```

      UU(K,3)=AKZ
      T(K+1)=XEND
      Z(K+1,1)=Z1
      Z(K+1,2)=Z2
30  CONTINUE
      WRITE(20,60)
60  FORMAT(/2X,'THREE JOINT ROBOT MANIPULATOR-FEEDBACK',
/      1X,'LINEARIZATION')
      WRITE(20,62)
62  FORMAT(2X,'TIME VARYING FEEDBACK GAINS.'//)
      WRITE(20,100)
100 FORMAT(///,9X,'TIME',12X,'R',13X,'X2',13X,'PHI',
/      13X,'X4',14X,'Z',13X,'X6'//)
      WRITE(20,105)(T(K),(AZ(K,I),I=1,N),K=1,45,5)
105 FORMAT(7(3X,F12.5))
      WRITE(20,120)
120 FORMAT(//8X,'TIME',11X,'FR',14X,'T',13X,'FZ',13X,'K1',13X,'K2'//)
      WRITE(20,125)(T(K+1),(UU(K,I),I=1,M),(Z(K+1,I),I=1,2),K=1,60)
125 FORMAT(6(3X,F12.5))
      GO TO 900
500 CONTINUE
      WRITE(20,550)
550 FORMAT(//4X,'X1',15X,'X2',15X,'X3',15X,'X4',
/      15X,'X5',15X,'X6'//)
      WRITE(20,600)(Y(I),I=1,N)
600 FORMAT(6(3X,F15.10))
      WRITE(20,*) (XEND,H,X,METH,MITER)
900 STOP
      END

```

```

SUBROUTINE FCNJ(N,X,Y,PD)
INTEGER N
REAL Y(N),PD(N,N),X
RETURN
END

```

```

SUBROUTINE FCN(N,X,Y,YPRIME)
INTEGER N
REAL Y(N),YPRIME(N),X,L
COMMON/A/B1,B2,B3,B4,AKR,AM1,AKZ
COMMON/B/Z1,Z2
L=1.
W1=0.5
W2=-0.6
W3=-0.2
ER=(W1-Y(1))**2+(W2-Y(3))**2+(W3-Y(5))**2
IF(ER-0.2)9,9,8
8  CC=1.01
GO TO 13
9  CC=1.15
13 C1=B2*Y(1)**2+(B4/4.)*(L-3.*Y(1))
F1=(B2*Y(1)-(3./8.)*B4)*(Y(4)**2/B1)
F2=(-2.*B2*Y(1)+(3./4.)*B4)*(Y(2)*Y(4)/C1)
C2=(B3/B1)*9.81*SIN(Y(3))
C3=(-3.*Y(1)+B4/2.)*(9.81*COS(Y(3))/C1)
5  E1=Z1*W1-Z1*Y(1)-Z2*Y(2)
E2=Z1*W2-Z1*Y(3)-Z2*Y(4)
E3=Z1*W3-Z1*Y(5)-Z2*Y(6)

```

```
AKR=B1*(C2-F1+E1)
AM1=C1*(-C3-F2+E2)
AKZ=B3*E3
20 IF(ABS(AM1)-400.)20,20,10
30 IF(ABS(AKR)-400.)30,30,10
40 IF(ABS(AKZ)-400.)40,40,10
60 IF(Z1-10000.)60,60,50
Z1=Z1*1.5
Z2=2.*CC*SQRT(Z1)
GO TO 5
10 Z1=Z1/1.5
Z2=2.*CC*SQRT(Z1)
E1=Z1*(W1-Y(1))-Z2*Y(2)
E2=Z1*(W2-Y(3))-Z2*Y(4)
E3=Z1*(W3-Y(5))-Z2*Y(6)
AKR=B1*(C2-F1+E1)
AM1=C1*(-C3-F2+E2)
AKZ=B3*E3
1 IF(ABS(AKZ)-400.)1,1,10
2 IF(ABS(AM1)-400.)2,2,10
50 IF(ABS(AKR)-400.)50,50,10
YPRIME(1)=Y(2)
YPRIME(2)=F1-C2+AKR/B1
YPRIME(3)=Y(4)
YPRIME(4)=F2+C3+(AM1/C1)
YPRIME(5)=Y(6)
YPRIME(6)=AKZ/B3
RETURN
END
```

APPENDIX B

```

PROGRAM RCB(INPUT,OUTPUT,TAPE10)
C*****
C      SIMULATION OF A THREE JOINT ROBOT MANIPULATOR-
C      COMPUTED TORQUE TECHNIQUE.
C      TIME VARYING FEEDBACK GAINS.
C*****

```

```

INTEGER N,METH,MITER,INDEX,IWK(6),IER,K
REAL    Y(6),WK(150),X,TOL,XEND,H,MR,ML,L
COMMON/A/B1,B2,B3,B4,AKR,AM1,AKZ
COMMON/B/Z1,Z2
DIMENSION AZ(120,10),UU(120,10),T(120),Z(120,10)
EXTERNAL FCN,FCNJ
N=6
M=3
X=0.0
Y(1)=1.
Y(2)=0.
Y(3)=0.
Y(4)=0.
Y(5)=-1.
Y(6)=0.
TOL=.00001
H=.0000001
METH=1
MITER=0
INDEX=1
MR=10.
ML=4.
L=1.
Z1=15.
Z2=2.*SQRT(Z1)
B1=ML+MR/4.
B2=ML+(3./4.)*MR
B3=ML+MR
B4=MR=L
DO 1 I=1,N
    AZ(1,I)=Y(I)
1  CONTINUE
T(1)=0.
Z(1,1)=Z1
Z(1,2)=Z2
DO 30 K=1,60
    XEND=.01*FLOAT(K)
    CALL DGEAR(N,FCN,FCNJ,X,H,Y,XEND,TOL,METH,
/      MITER,INDEX,IWK,WK,IER)
    IF(IER.GT.128)GO TO 500
    DO 50 I=1,N
        AZ(K+1,I)=Y(I)
50  CONTINUE
    UU(K,1)=AKR
    UU(K,2)=AM1

```

```

      UU(K,3)=AKZ
      T(K+1)=XEND
      Z(K+1,1)=Z1
      Z(K+1,2)=Z2
30  CONTINUE
      WRITE(10,60)
60  FORMAT(/2X,'THREE JOINT ROBOT MANIPULATOR-COMPUTED',
/      1X,'TORQUE TECHNIQUE')
      WRITE(10,62)
62  FORMAT(2X,'TIME VARYING FEEDBACK GAINS. '//)
      WRITE(10,100)
100 FORMAT(///,8X,'TIME',12X,'R',13X,'X2',13X,'PHI',
/      13X,'X4',14X,'Z',13X,'X6'//)
      WRITE(10,105)(T(K),(AZ(K,I),I=1,N),K=1,60)
105 FORMAT(7(3X,F12.6))
      WRITE(10,120)
120 FORMAT(//8X,'TIME',11X,'FR',14X,'T',13X,'FZ',13X,'K1',13X,'K2'//)
      WRITE(10,125)(T(K+1),(UU(K,I),I=1,M),(Z(K+1,I),I=1,2),K=1,60)
125 FORMAT(6(3X,F12.5))
      GO TO 900
500 CONTINUE
      WRITE(10,550)
550 FORMAT(//4X,'X1',15X,'X2',15X,'X3',15X,'X4',
/      15X,'X5',15X,'X6'//)
      WRITE(10,600)(Y(I),I=1,N)
600 FORMAT(6(3X,F15.10))
      WRITE(10,*) (XEND,H,X,METH,MITER)
900 STOP
      END

```

```

SUBROUTINE FCNJ(N,X,Y,PD)
  INTEGER N
  REAL Y(N),PD(N,N),X
  RETURN
  END

```

```

SUBROUTINE FCN(N,X,Y,YPRIME)
  INTEGER N
  REAL Y(N),YPRIME(N),X,L
  COMMON/A/B1,B2,B3,B4,AKR,AM1,AKZ
  COMMON/B/Z1,Z2
  L=1.
  X71=1.
  X81=0.5
  X91=-0.3
  X72=0.5
  X32=0.5
  X92=0.5
  X73=0.2
  X83=0.2
  X93=0.2
  C1=B2*Y(1)**2+(B4/4.)*(L-3.*Y(1))
  F1=(B2*Y(1)-(3./8.)*B4)*(Y(4)**2/B1)
  F2=(-2.*B2*Y(1)+(3./4.)*B4)*(Y(2)*Y(4)/C1)
  C2=(B3/B1)*9.81*SIN(Y(3))
  C3=(-B3*Y(1)+B4/2.)*(9.81*COS(Y(3))/C1)
5  E1=B1*(X73+Z1*(X71-Y(1))+Z2*(X72-Y(2)))
  AKR=E1*(-B2*Y(1)+(3./8.)*B4)*Y(4)**2+B3*9.81*SIN(Y(3))

```

```
E2=C1*(X83+Z1*(X81-Y(3))+Z2*(X82-Y(4)))
E3=E2+(2.*B2*Y(1)-(3./4.)*B4)*Y(2)*Y(4)
AM1=E3+(B3*Y(1)-B4/2.)*9.81*COS(Y(3))
AKZ=B3*(X93+Z1*(X91-Y(5))+Z2*(X92-Y(6)))
IF(ABS(AM1)-5000.)20,20,10
20 IF(ABS(AKR)-5000.)30,30,10
30 IF(ABS(AKZ)-5000.)40,40,10
40 IF(Z1-50000.)60,60,50
60 Z1=Z1*2.3
Z2=2.*SQRT(Z1)
GO TO 5
10 Z1=Z1/2.3
Z2=2.*SQRT(Z1)
E1=B1*(X73+Z1*(X71-Y(1))+Z2*(X72-Y(2)))
AKR=E1+(-B2*Y(1)+(3./8.)*B4)*Y(4)**2+B3*9.81*SIN(Y(3))
E2=C1*(X83+Z1*(X81-Y(3))+Z2*(X82-Y(4)))
E3=E2+(2.*B2*Y(1)-(3./4.)*B4)*Y(2)*Y(4)
AM1=E3+(B3*Y(1)-B4/2.)*9.81*COS(Y(3))
AKZ=B3*(X93+Z1*(X91-Y(5))+Z2*(X92-Y(6)))
50 YPRIME(1)=Y(2)
YPRIME(2)=F1-C2+AKR/B1
YPRIME(3)=Y(4)
YPRIME(4)=F2+C3+(AM1/C1)
YPRIME(5)=Y(6)
YPRIME(6)=AKZ/B3
RETURN
END
```

Part 2

Stability of Time-Varying System

In robot control, variable effective inertia and gravity loading effects suggest the imperative need for time-varying models. It has been shown [1] that proper use of time-varying controller can produce fast robot manipulator motion without causing undesirable overshoot. Most of currently available methods of controller design require precise parameter values of the plant, which are often impossible or impractical to have in practice. In recent years a new control design philosophy has emerged. This is the intelligent control [2]. An intelligent control is capable of updating its control strategies through learning. It is operating on heuristic as well as analytic reasoning. It employs both quantitative and qualitative information in its decision making processes. In fact, very often the qualitative information is placed higher than the quantitative information in its ruled-based decision hierarchy.

Stability is the first requirement in any satisfactory control action. stability criteria for linear time-invariant systems are widely available, but explicit and practical stability criteria for time-varying systems are still lacking. However, explicit stability criteria for certain special classes [3] of linear time-varying systems are available. An intelligent control is inherently a time-varying control. In the following we shall examine the stability of a special class of periodically time-varying system. The effect of rate of parameter variations on stability will be examined in detail. We believe the results obtained here will provide the qualitative information that may provide a general guideline for designing an intelligent control of robot manipulation. As will be seen in subsequent analysis rate of parameter variations appears to be most critical to stability for frequency in a band centering around the "resonant frequency" of its constant nominal system.

Consider a linear time-varying system

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bu(t) \quad (1)$$

the stability of $A(t)$ in general is still an open problem. However stability for special $A(t)$ can be precisely determined. Results obtained from such analysis may provide useful qualitative information in the design of intelligent control of robot manipulator. To deal with a manageable problem at this point, we shall assume that $A(t)$ can be separated into

$$A(t) = A_0 + A_1(t)$$

where A_0 is a constant nominal part and $A_1(t)$ is a time-varying part that represents varying parameters in the system. It is further assumed that associated with $A_1(t)$ there is a parameter ω that governs the rate of parameter variations in the system. The parameter ω can be viewed as a quantity that specifies how fast the time-varying controller is changing or how fast the variable inertia or gravity loading of robot manipulator may change in its operation. The important question of interest

is how ω will affect the stable operation of the system, i.e. how ω affects the stability of the system. To gain insight into this problem, we shall examine the following special system. Consider the system (1) with $A(t)$ being

$$A(t) = \begin{bmatrix} \alpha + \gamma \cos \omega t + \delta \sin \omega t & \beta + \delta \cos \omega t - \gamma \sin \omega t \\ -\beta + \delta \cos \omega t - \gamma \sin \omega t & \alpha - \gamma \cos \omega t - \delta \sin \omega t \end{bmatrix} \quad (2)$$

Equation (2) above can be written as

$$A(t) = A_0 + A_1(t) \quad (3)$$

with

$$A_0 = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \quad (4a)$$

and

$$A_1 = \begin{bmatrix} \gamma \cos \omega t + \delta \sin \omega t & \delta \cos \omega t - \gamma \sin \omega t \\ \delta \cos \omega t - \gamma \sin \omega t & -\gamma \cos \omega t - \delta \sin \omega t \end{bmatrix} \quad (4b)$$

where A_0 represents the constant nominal part and $A_1(t)$ represents the time-varying part due to parameter variations. It should be commented that $A_1(t)$ can also be written as

$$A_1(t) = \begin{pmatrix} \gamma & \delta \\ \delta & -\gamma \end{pmatrix} \begin{pmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{pmatrix} \\ \triangleq BK(t) \quad (5a)$$

or

$$A_1(t) = \begin{pmatrix} \cos \omega t & \sin \omega t \\ -\sin \omega t & \cos \omega t \end{pmatrix} \begin{pmatrix} \gamma & \delta \\ \delta & -\gamma \end{pmatrix} \\ \triangleq L(t)C \quad (5b)$$

These interpretations permit us to examine the time-varying effects of control or feedback. It can be shown [4] that $A(t)$ in (2) is stable if and only if

$$\alpha < 0$$

and

$$\alpha^2 + \beta^2 - \gamma^2 - \delta^2 + \frac{\omega^2}{4} - \beta\omega > 0$$

It will be interesting to see the physical significance of those conditions. Note that the constant nominal system A_0 is stable if and only if $\alpha < 0$. Therefore for the time-varying system $A(t)$ to be stable, it is necessary that the constant nominal system be stable. However such a condition will not be in general sufficient to ensure the stability under parameter variations. It is intuitively clear that the rate

of variations in the parameters should affect the overall stability. Here we like to examine the effect of ω on stability in more detail.

The effect of ω on stability can be examined through the following condition:

$$f(\omega) \equiv \left(\frac{\omega}{2} - \beta\right)^2 + \alpha^2 - (\gamma^2 + \delta^2) > 0 \quad (6)$$

It is easy to see that if the constant nominal system A_0 has sufficient damping to suppress the perturbation induced by parameter variations, namely if

$$\alpha^2 > \gamma^2 + \delta^2$$

then the above time-varying system is stable for all ω . In other words, if the magnitude of parameter variations is not large enough to upset the stability of the constant nominal system, then the rate of parameter variation has no bearing on the overall stability.

It can also be seen that if the magnitude of variations is sufficient to cause instability i.e. if

$$\gamma^2 + \delta^2 > \alpha^2$$

great attention should be paid to the rate of variations. It is found that stability is determined by a critical frequency band $[\omega_1, \omega_2]$ (to be called the instability zone), where

$$\omega_1 = 2 \left(\beta - \sqrt{\gamma^2 + \delta^2 - \alpha^2} \right) \quad (7a)$$

and

$$\omega_2 = 2 \left(\beta + \sqrt{\gamma^2 + \delta^2 - \alpha^2} \right) \quad (7b)$$

The system is unstable if $\omega \in [\omega_1, \omega_2]$ and it is stable if $\omega \notin [\omega_1, \omega_2]$. Note that the instability zone is centered at 2β , twice the damped frequency of the constant nominal system, rather than centering at its undamped natural frequency given by $\omega_n = \sqrt{\alpha^2 + \beta^2}$, as intuition may suggest.

Some useful observations concerning stability of time-varying system are summarized below:

1. Magnitude of variations is found to be more significant than the rate of variations in affecting the system stability.
2. For a time-varying system to be stable, very often it is necessary to have a stable constant nominal system.
3. If the magnitude of variations is large enough to upset the stability, the system tends to have the worst destabilizing effect when the rate of variation is in or near the instability frequency band centred at twice the damped frequency of its constant nominal system. In other words, it is important to avoid exciting possible resonance. It is also noted that very fast variation (when ω is large) or very slow variation (when ω is small) tend to offer better chance of avoiding instability.

of variations in the parameters should affect the overall stability. Here we like to examine the effect of ω on stability in more detail.

The effect of ω on stability can be examined through the following condition:

$$f(\omega) \equiv \left(\frac{\omega}{2} - \beta\right)^2 + \alpha^2 - (\gamma^2 + \delta^2) > 0 \quad (6)$$

It is easy to see that if the constant nominal system A_0 has sufficient damping to suppress the perturbation induced by parameter variations, namely if

$$\alpha^2 > \gamma^2 + \delta^2$$

then the above time-varying system is stable for all ω . In other words, if the magnitude of parameter variations is not large enough to upset the stability of the constant nominal system, then the rate of parameter variation has no bearing on the overall stability.

It can also be seen that if the magnitude of variations is sufficient to cause instability i.e. if

$$\gamma^2 + \delta^2 > \alpha^2$$

great attention should be paid to the rate of variations. It is found that stability is determined by a critical frequency band $[\omega_1, \omega_2]$ (to be called the instability zone), where

$$\omega_1 = 2 \left(\beta - \sqrt{\gamma^2 + \delta^2 - \alpha^2} \right) \quad (7a)$$

and

$$\omega_2 = 2 \left(\beta + \sqrt{\gamma^2 + \delta^2 - \alpha^2} \right) \quad (7b)$$

The system is unstable if $\omega \in [\omega_1, \omega_2]$ and it is stable if $\omega \notin [\omega_1, \omega_2]$. Note that the instability zone is centered at 2β , twice the damped frequency of the constant nominal system, rather than centering at its undamped natural frequency given by $\omega_n = \sqrt{\alpha^2 + \beta^2}$, as intuition may suggest.

Some useful observations concerning stability of time-varying system are summarized below:

1. Magnitude of variations is found to be more significant than the rate of variations in affecting the system stability.
2. For a time-varying system to be stable, very often it is necessary to have a stable constant nominal system.
3. If the magnitude of variations is large enough to upset the stability, the system tends to have the worst destabilizing effect when the rate of variation is in or near the instability frequency band centred at twice the damped frequency of its constant nominal system. In other words, it is important to avoid exciting possible resonance. It is also noted that very fast variation (when ω is large) or very slow variation (when ω is small) tend to offer better chance of avoiding instability.

Part 3

Learning Controller Design

Contents

	Page
Abstract.....	1
Outline.....	1
1.0 Betterment Processes.....	1
1.1 Background.....	2
1.2 C^0 -Type Betterment Process.....	3
1.3 C^1 And Mixed Type Betterment.....	3
1.4 Remarks And Discussion.....	4
Example 1.1.....	6
2.0 Betterment Process Based On The Decomposition Of Functions.....	6
2.1 Theoretical Background.....	6
2.2 Developement Of The Scheme.....	8
2.2.1 Development Of The Iteration Method.....	9
2.3 Remarks and Discussion.....	12
Example 2.1.....	13
3.0 Comparison Of Learning Algorithms.....	13
3.1 An Extended Algorithm.....	14
3.2 Generalization Of The Algorithm.....	15
Example 3.1.....	17
3.4 Remarks And Discussion.....	18
Bibliography.....	18
Figures.....	20

Abstract

The possibility of controlling dynamical systems under incomplete and even very small a priori information is based on the application of adaptation and learning in automatic systems which reduces initial uncertainty by using the information obtained during the process of control. It goes without saying that adaptive techniques in control (but also in filtering and prediction) have been extensively studied for over a decade, and not long ago a rigorous and comprehensive theory of convergence of adaptive algorithms has emerged [2]. Also in practice numerous successful applications have been reported ([1], [2]). At the same time little attention has been given to "learning" and "self learning" in the world of control engineering until recently when S. Arimoto and his co-workers [3] proposed a learning control method called betterment process based on a simple iterative algorithm, which was successfully applied to the motion control of robots [4]. A closer look at the betterment process may lead to other alternatives and perhaps to a unified (generalized) method. The development of such learning control methods along with possible applications and problems for research in learning control form the basis of this thesis. Before presenting the thesis outline it is important to notice that the terms "adaptation" & "learning" do not have a unique interpretation and usually their definitions vary from one author to another or even from one technical paper to another [1]. It is certainly not the purpose of the author to get involve in the technical terminology but since "learning" seems to be easier to understand in the context of this thesis, it will be therefore used exclusively (unless otherwise stated).

Outline of the first part

This part of the thesis consists of the following four sections. In the first section the betterment theory is briefly reviewed : mathematical background, main theorems, and different schemes are introduced. Some basic remarks and notes end this section. Section two contains the theoretical basis of learning methods based on function decomposition, to do that some results from linear algebra and functional analysis are needed. In the third section a more powerful and general algorithm is developed. Remarks and discussions concerning the limitations and applications of the algorithm conclude this section. In the last section of this part of the thesis the extension of the decomposition techniques to a class of nonlinear systems is investigated, it is also shown that a combination in a certain way of the betterment algorithm and the decomposition techniques can yield to a faster learning .

1.0 Betterment Processes

In this section an algorithm called betterment process for linear systems is introduced. This algorithm updates the control input based on the previous operation data and "betters" the performance of the next operation in a certain sense, provided a desired output is given. In this section three types of learning control scheme are discussed.

1.1 Background

Consider the linear time invariant continuous system described by the following state equations:

$$\dot{X}(t) = AX(t) + Bu(t) \quad (1a)$$

$$y(t) = CX(t) \quad (1b)$$

where t represents the time in the interval $[0, T]$. If we assume that for each run the initial state $X_k(0)$ is the same fixed state x^o then in the k -th run the output of the system(1) is

$$\begin{aligned} y_k(t) &= Ce^{-At}X_k(0) + \int_0^t Ce^{A(t-\tau)}Bu_k(\tau)d\tau \\ &= Ce^{At}x^o + \int_0^t Ce^{A(t-\tau)}Bu_k(\tau)d\tau \end{aligned} \quad (1.2a)$$

$$= g(t) + \int_0^t h(t-\tau)u_k(\tau)d\tau \quad (1.2b)$$

From (1.2) it is clear that $g(t)$ and $h(t)$ are the same for each run in the interval $[0, T]$. That is

$$g(t) = Ce^{At}x^o \quad (1.3a)$$

$$h(t) = Ce^{At}B \quad (1.3b)$$

Definition 1. (see [3],[5]) A linear time invariant system described by (1.2) is said to be strictly positive, if for any $T > 0$ and any $u(t)$, t is in $[0, T]$ the following inequality is satisfied with some constant $\alpha > 0$

$$\int_0^T \int_0^t u^T(t) h(t-\tau) u(\tau) d\tau dt \geq \alpha \int_0^T u^T(t) u(t) dt \quad (1.4)$$

Definition 2. Given a vector valued function $u(t)$, t in the interval $[0, T]$ then the L2-norm of $u(t)$ is defined by

$$\|u\| = \left[\int_0^T u^T(t) u(t) dt \right]^{1/2} \quad (1.5)$$

Definition 3. The spectral radius γ_o of a matrix A is defined as

$$\gamma_o = \rho\{A\} = \max_{\lambda \in \sigma\{A\}} |\lambda|$$

where $\sigma\{A\}$ represents the set of the eigenvalues of the matrix A .

Definition 4. A rational transfer function matrix $H(s)$ is said to be proper if

$$\lim_{s \rightarrow -\infty} H(s) < \infty$$

and strictly proper if

$$\lim_{s \rightarrow \infty} H(s) = 0 \quad (\text{zero matrix})$$

In the scalar case, a transfer function is proper if the degree of the numerator polynomial is less than or equal to the degree of the denominator polynomial.

1.2 C^0 -Type Betterment Process

Consider the linear time invariant continuous system described by (1) where the input vector $u(t)$, and the output vector $y(t)$ have the same dimension. Also suppose that a desired output vector $y_d(t)$ is given over the interval $[0, T]$, then the C^0 -type betterment process is defined by

$$u_{k+1}(t) = u_k(t) + \Gamma e_k(t) \quad (1.6)$$

where

$$e_k(t) = y_d(t) - y_k(t) \quad (1.7)$$

and Γ is an $m \times m$ constant gain matrix to be defined. $y_k(t)$ and $u_k(t)$ are respectively the m -dimensional output and input of the system (1) in the k -th run.

Theorem 1. Suppose that the linear time invariant continuous system (1) is strictly positive and

$$\Gamma = \gamma I \quad (1.8)$$

I is the $m \times m$ identity matrix and γ is a sufficiently small and positive constant. Then the C^0 -type betterment process is convergent in the sense that

$$\|e_{k+1}\| \leq \rho \|e_k\| \quad (1.9)$$

where

$$0 \leq \rho < 1$$

The proof of this theorem can be found in [3].

1.3 C^1 And Mixed Type Betterment

The C^1 -type betterment is described by the following simple iterative rule of input modification:

$$u_{k+1}(t) = u_k(t) + \Gamma \frac{d}{dt} \{y_d(t) - y_k(t)\} \quad (1.10)$$

Here also, $u_k(t)$ and $y_k(t)$ are the m -dimensional system input and system output respectively (the system under test is system (1)). The constant Γ is an $m \times m$ constant matrix called the "gain matrix"

Theorem 2. The C^1 -type betterment process defined by (1.10) converges in the sense that as $k \rightarrow \infty$ $y_k(t) \rightarrow y_d(t)$ uniformly in t over $[0, T]$ if the following conditions are satisfied:

1:

$$\|I_m - CB\Gamma\|_\infty < 1$$

where $\|\cdot\|_\infty$ is the matrix norm induced by vector norm v and

$$\|v\|_\infty = \max_{1 \leq i \leq m} |v_i|$$

2:

$$u_o(t), y_d(t) \text{ are in } C^1[0, T]$$

3:

$$y_d(0) = Cx^o$$

The proof of theorem 2 can be found in [3].

Obviously, since the parameters of the system (1) are assumed to be unknown condition 1 does not help in choosing the gain matrix Γ . In comparison with the above two types of betterment, it is also useful—as will be shown in the next subsection—to mention the so called mixed-type betterment process defined by:

$$u_{k+1}(t) = u_k(t) + \left(\Phi + \Gamma \frac{d}{dt} \right) e_k(t). \quad (1.11)$$

1.4 Remarks And Discussion

In the above subsections different function norms have been chosen in evaluating the performance of different types of betterment processes. For the C^1 -type betterment the uniform norm $\|\cdot\|_\lambda$ becomes inadequate when the considered time interval $[0, T]$ expands, because the constant λ becomes too large. Although it can be shown that for an asymptotically stable linear system a fixed value for λ is permitted for any expansion of the interval $[0, T]$, it may be more suitable to choose some other kind of function norms, especially when the desired output $y_d(t)$ is defined over the semi-infinite interval $[0, \infty)$. For example, if we consider the L-2 norm for the C^1 -type betterment process and the desired output $y_d(t)$ is defined over the semi-infinite interval $[0, \infty)$ then it can be shown that

$$\dot{e}_{k+1}(t) = (I_m - CB\Gamma)\dot{e}_k(t) - \int_0^t CAe^{A(t-\tau)}B\Gamma\dot{e}_k(\tau)d\tau \quad (1.12)$$

provided that all eigenvalues of matrix A have negative real parts. Now if we take the Laplace transform of the above equation and denote

$$E_{k+1}(s) = \mathcal{L}\{\dot{e}_{k+1}(t)\}$$

$$E_k(s) = \mathcal{L}\{\dot{e}_k(t)\} \quad (1.13)$$

$$H(s) = C(sI - A)^{-1}B$$

then we get

$$E_{k+1}(s) = (I_m - sH(s)\Gamma)E_k(s) \quad (1.14)$$

By putting

$$\gamma = \sup_{\omega} \rho\{I_m - j\omega H(j\omega)\Gamma\} \quad (1.15)$$

where $\rho\{X\}$ is the spectral radius of the matrix X and $\|E\|$ is defined for a vector valued function $E = (E_1, \dots, E_m)^T$ by

$$\|E\|^2 = \int_{-\infty}^{+\infty} \sum_{i=1}^m |E_i(j\omega)|^2 d\omega \quad (1.16)$$

we get

$$\|E_{k+1}\| < \gamma \|E_k\| \quad (1.17)$$

which implies—according to Parseval's equality—the following inequality

$$\|\dot{e}_{k+1}\| < \gamma \|\dot{e}_k\| \quad (1.18)$$

Therefore we can conclude that the betterment process converges in the L2-norm sense if γ is less than unity. However this is impossible unless the denominator of $H(s)$ vanishes at $s=0$ as one can see from equation (1.15). This same remark is also relevant for the C^0 - type betterment process. In this case and considering the frequency domain (as was done above) the convergence of the process is assured if

$$\gamma_0 = \sup_{\omega} \rho\{I_m - H(j\omega)\Phi\} < 1 \quad (1.19)$$

However this condition is not always satisfied for linear causal systems with a proper transfer function matrix, because in general for such systems $H(s) \rightarrow 0$ as $s \rightarrow \infty$. In view of these arguments the convergence of the mixed type betterment process can be assured if one can choose the appropriate matrices Γ and Φ such that

$$\gamma = \sup_{\omega} \rho\{I_m - (\Phi + \Gamma j\omega)H(j\omega)\} < 1. \quad (1.20)$$

which leads to the following question : What is the class of linear dynamical systems for which such matrices Φ and Γ exist ? The study of this question and the question of using other types of betterment process with different function norms are interesting subjects of research.

Before ending this section it is necessary to mention that for the above algorithms it is sufficient (but not necessary) to reset the initial state in each run however it is necessary that for each run (1.3a) holds. In this case condition 3 of theorem 2 changes to

$$y_d(0) = y_k(0) \text{ for all } k$$

Example 1.1

To see the applicability of the scheme presented in this section, consider the following system:

$$\begin{aligned}\dot{X} &= AX + Bu \\ y &= CX\end{aligned}\tag{1.21}$$

Where

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [0 \quad 1]$$

and the desired output is

$$y_d(t) = 12t^2(1-t)\tag{1.22}$$

$t \in [0, 1]$. The C^1 -type betterment has been chosen for this example. Since $CB = 1$ the assumption

$$\|I - \Gamma CB\|_\infty < 1\tag{1.23}$$

is now given by:

$$|1 - \Gamma| < 1\tag{1.24}$$

In this particular example we choose the constant Γ to be

$$\Gamma = 1\tag{1.25}$$

Figure (1.1) shows that in a few iterations ($k = 4$) the desired output $y_d(t)$ is achieved. Finally, it should be noted that it is impossible to choose $y(t) = x_1(t)$ as output of the system (1.21) because in this case $CB = 0$ and there exist no Γ for which (1.23) can be satisfied.

2.0 Betterment Process Based On The Decomposition Of Functions

In this section an iterative method for betterment process is developed. The desired output $y_d(t)$ is a continuous function defined over the interval $[0, T]$ where $T < \infty$ is a given constant. The desired output $y_d(t)$ of a single-input single-output linear time invariant continuous system is expressed as a linear combination of functions from a complete orthonormal set given a priori. The iteration method is then applied on such decomposition.

2.1 Theoretical Background

The following definitions and theorems are needed in the development of the scheme presented in this part.

Definition 5. A set of functions $\phi_i(t)$ $i = 1, 2, \dots$ defined on an interval $[t_1, t_2]$ is said to be orthogonal if

$$\int_{t_1}^{t_2} \phi_i(t) \phi_j(t) dt = \begin{cases} 0, & i \neq j \\ k_i, & i = j \end{cases} \quad (2.1)$$

and orthonormal if in (2.1) the constant $k_i = 1$

Definition 6. The orthonormal set $\{\phi_1(t), \phi_2(t), \dots\}$ is said to be complete if and only if it is not a subset of a larger orthonormal set.

Let the approximation of $f(t)$ on $[t_1, t_2]$ for a given set of n orthogonal functions $\phi_1(t), \phi_2(t), \dots, \phi_n(t)$ be

$$f(t) \approx c_1 \phi_1(t) + c_2 \phi_2(t) + \dots + c_n \phi_n(t) \quad (2.2)$$

the mean square error (MSE) between the true value of $f(t)$ and the approximation $\sum_{i=1}^n c_i \phi_i(t)$ is given by

$$MSE = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left[f(t) - \sum_{i=1}^n c_i \phi_i(t) \right]^2 dt \quad (2.3)$$

It can be shown that the best approximation of $f(t)$ on $[t_1, t_2]$ in the sense of minimum mean square error is when $c_i, i = 1, 2, \dots, n$ are chosen as follows

$$c_i = \gamma_i / k_i \quad i = 1, 2, \dots, n \quad (2.4)$$

where

$$\begin{aligned} \gamma_i &= \int_{t_1}^{t_2} f(t) \phi_i(t) dt \\ k_i &= \int_{t_1}^{t_2} \phi_i^2(t) dt \end{aligned} \quad (2.5)$$

Definition 7. The function $\tilde{f}(t) = \tilde{f}(\epsilon, N, \Phi)$ defined in the interval $[t_1, t_2]$ is said to be an ϵ -approximation of a given function $f(t)$ on the interval $[t_1, t_2]$, with respect to a chosen set of orthogonal functions described by the vector valued function:

$$\Phi = [\phi_1(t), \phi_2(t), \phi_3(t), \dots, \phi_n(t)]^T$$

if for a given positive ϵ there exists a number N such that for all t in $[t_1, t_2]$

$$\|f - \tilde{f}\| \leq \epsilon \text{ for } n \geq N$$

lemma 1. Let the real valued function $f(t)$ be square-integrable on the interval $[t_1, t_2]$ then the ϵ -approximation of $f(t)$ with respect to Φ is given by

$$\tilde{f}(t) = \sum_{i=1}^N c_i \phi_i(t)$$

where c_i are defined by (2.4) and (2.5)

There are many sets of functions that can be used to represent a function on an interval $[t_1, t_2]$. For examples the sets of $\{\cos(n\omega t)\}$, $\{\sin(n\omega t)\}$, the set of Walsh functions and the set of Legendre functions defined on $[-1, 1]$ by

$$\phi_0(t) = 1/\sqrt{2}, \phi_1(t) = t\sqrt{3/2}, \dots, \phi_n(t) = \sqrt{\frac{2n+1}{2}} p_n(t)$$

where the Legendre polynomials $p_n(t)$ are generated by the formula

$$p_n(t) = \frac{1}{2^n n!} \frac{d^n}{dt^n} (t^2 - 1)^n$$

In this part—as previously—the L2 norm will be used. For that reason the following two theorems (see [6] for proof) will be important in establishing the relation between the concept of multiplier (linear, continuous time invariant operator) and the concept of transfer functions of the space $K(0)$ which contains all complex valued functions $s \rightarrow H(s)$ of the complex variable s , and it is bounded and holomorphic in the open right half plane $\{s : \text{Re}(s) > 0\}$. $K(0)$ is normed algebra under the pointwise multiplication of functions and under the norm

$$\|H\| = \sup_{\text{Re}(s) > 0} |H(s)|$$

Theorem 3. (*L2-representation theorem*) Let $M(L2)$ represent the algebra of all multipliers in L2. Then there exists an isomorphism of rings such that to each A in $M(L2)$ there is assigned a transfer function H in $K(0)$ satisfying

(i)

$$\mathcal{L}(Af)(s) = H(s)\mathcal{L}(f)(s)$$

for all f in L2 and all $\text{Re}(s) > 0$

(ii)

$$\|A\| = \|H\|$$

For completeness the L^{∞} -representation theorem is also given.

Theorem 4. (*L^{∞} -representation theorem*) The ring of all multipliers in L^{∞} is isomorphic with the ring of all $n \times n$ matrices over $K(0)$, in such a way that to each A of $M(L^{\infty})$ there is assigned \hat{A} in $K(0)^{n \times n}$, called the matrix transfer function of A such that

$$(\mathcal{L}Af)(s) = \hat{A}(s)(\mathcal{L}f)(s)$$

for all f in L^{∞} and all $\text{Re}(s) > 0$

The proofs of the above two theorems are found in [6].

2.2 Development Of The Scheme

2.2.1 Development Of The Iteration Method

Let the set $\{\phi_1(t), \phi_2(t), \dots, \phi_n(t)\}$ represent a set of orthonormal functions defined on $[0, T]$ where $T > 0$ is a given constant. Consider also the vector valued function

$$\Phi = [\phi_1(t), \phi_2(t), \phi_3(t), \dots, \phi_n(t)]^T$$

Let $\tilde{y}_d(t)$ be the ϵ -approximation of the desired output of a single-input single-output time invariant continuous system and defined as

$$\tilde{y}_d(t) = \sum_{i=1}^n \alpha_i^d \phi_i(t) \quad (2.6)$$

and let the input and the output of the k -th run be respectively ϵ -approximated by

$$\tilde{u}_k(t) = \sum_{i=1}^n \beta_i^k \phi_i(t) \quad (2.7)$$

$$\tilde{y}_k(t) = \sum_{i=1}^n \alpha_i^k \phi_i(t) \quad (2.8)$$

From (2.6) and (2.8) we get

$$\tilde{e}_k(t) = \tilde{y}_d(t) - \tilde{y}_k(t) = \sum_{i=1}^n (\alpha_i^d - \alpha_i^k) \phi_i(t) \quad (2.9)$$

In this case we can use the following vector representation

$$\tilde{u}_k(t) = [\beta_1^k, \beta_2^k, \dots, \beta_n^k] \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \phi_n(t) \end{bmatrix} = \underline{\beta}^T(k) \Phi \quad (2.10)$$

where $\underline{\beta}(k) = [\beta_1^k, \beta_2^k, \dots, \beta_n^k]^T$ Similarly we have

$$\tilde{y}_k(t) = \underline{\alpha}^T(k) \Phi, \quad \underline{\alpha}(k) = [\alpha_1^k, \alpha_2^k, \dots, \alpha_n^k]^T \quad (2.11)$$

and

$$\tilde{y}_d(t) = \underline{d}^T \Phi, \quad \underline{d} = [\alpha_1^d, \alpha_2^d, \dots, \alpha_n^d]^T \quad (2.12)$$

If we consider the C^0 -type betterment we get for t in $[0, T]$

$$\tilde{u}_{k+1}(t) = \sum_{i=1}^n \beta_i^{k+1} \phi_i(t) \quad (2.13)$$

where

$$\beta_i^{k+1} = \beta_i^k + \gamma(\alpha_i^d - \alpha_i^k) \quad (2.14)$$

In vector notation equation (2.14) is equivalent to

$$\underline{\beta}(k+1) = \underline{\beta}(k) + \gamma(\underline{d} - \underline{\alpha}(k)) \quad (2.15)$$

Now consider that $y_k(t)$ and $u_k(t)$ are related by a linear operator L as follows :

$$y_k(t) = L(u_k(t)) \quad (2.16)$$

Therefore

$$\tilde{y}_k(t) = L\left(\sum_{i=1}^n \beta_i^k \phi_i(t)\right) \quad (2.17)$$

But since L is linear then

$$\tilde{y}_k(t) = \sum_{i=1}^n \beta_i^k L(\phi_i(t)) \quad (2.18)$$

In the basis $\{\phi_1(t), \dots, \phi_n(t)\}$, $L(\phi_i(t))$ can be written as

$$L(\phi_i(t)) = \sum_{j=1}^n p_{ij} \phi_j(t) \quad (2.19)$$

or in vector form

$$L(\Phi) = P\Phi \quad (2.20)$$

where $P = [p_{ij}]$ is a $n \times n$ constant matrix. Therefore, in vector form (2.18) can be written as

$$\tilde{y}_k(t) = \beta^T(k) P \Phi \quad (2.21)$$

Using (2.21) and (2.11) we have

$$\underline{\alpha}(k) = P^T \underline{\beta}(k) \quad (2.22)$$

Plugging (2.22) in (2.14) we get

$$\underline{\beta}(k+1) = \underline{\beta}(k) + \gamma(\underline{d} - P^T \underline{\beta}(k)) \quad (2.23)$$

or

$$\underline{\beta}(k+1) = (I - \gamma P^T) \underline{\beta}(k) + \gamma \underline{d} \quad (2.24)$$

The convergence of Equation (2.24) and the choice of γ are stated in the following theorem.

Theorem 5.

Let $y_d(t)$ be a given desired trajectory defined over a time interval $[t_1, t_2]$ and let $e_k(t)$ be as defined in (1.7) and $\tilde{e}_k(t)$ be its ϵ -approximation with respect to Φ . Also let

$$L(\Phi) = P\Phi \quad (2.25)$$

where P is a $N \times N$ constant matrix, and L is the linear operator representing the system. Then

$$\|\tilde{e}_k\| \rightarrow 0 \text{ as } k \rightarrow \infty \quad (2.26)$$

if and only if

$$(\text{tr}(P))^2 > (N-1)\|P\|_F^2 \quad (2.27)$$

where $\|P\|_F^2$ and $\text{tr}P$ are the Forbenious norm and the trace of P respectively.

Proof:

From (2.8) we have

$$\tilde{e}_{k+1}(t) = \sum_{i=1}^n (\alpha_i^d - \alpha_i^{k+1}) \phi_i(t) \quad (2.28)$$

By letting

$$\underline{\varphi}(k+1) = [\alpha_1^d - \alpha_1^{k+1}, \dots, \alpha_n^d - \alpha_n^{k+1}]^T \quad (2.29)$$

we get

$$\tilde{e}_{k+1}(t) = \underline{\varphi}^T(k+1)\Phi \quad (2.30)$$

Also (2.29) can be written as

$$\underline{\varphi}(k+1) = \underline{d} - \underline{\alpha}(k+1) \quad (2.31)$$

Now using (2.22) and (2.24) we get from (2.31)

$$\underline{\varphi}(k+1) = \underline{d} - P^T \underline{\beta}(k) - \gamma P^T (\underline{d} - \underline{\alpha}(k)) \quad (2.32)$$

or

$$\underline{\varphi}(k+1) = (I - \gamma P^T) \underline{\varphi}(k) \quad (2.33)$$

Without loss of generality it is assumed that the basis is orthonormal therefore we conclude using the definition of L-2 norm

$$\|\tilde{e}_{k+1}\| = \left[\int_0^T \tilde{e}_{k+1}^2(t) dt \right]^{1/2}$$

that

$$\|\tilde{e}_k\| = \left[\sum_{i=1}^n (\varphi_i^{k+1})^2 \right]^{1/2} = \|\underline{\varphi}(k+1)\| \quad (2.34)$$

To simplify the notation let

$$F = I - \gamma P^T \quad (2.35)$$

(2.33), (2.34) and (2.35) imply that

$$\|\tilde{e}_{k+1}\| = \|F\underline{\varphi}(k)\| \quad (2.36)$$

Consider the Frobenius norm of the matrix F which is equal by definition to

$$\|F\|_F = [\text{tr}(F^T F)]^{1/2} \quad (2.37)$$

then it can be shown that for any v, w in R^n we have

$$\|Fv\| \leq \|F\|_F \|v\| \quad (2.38)$$

in particular

$$\|F\underline{\varphi}(k)\| \leq \|F\|_F \|\underline{\varphi}(k)\| \quad (2.39)$$

By letting $\|F\|_F = \rho$ and by choosing γ such that $0 \leq \rho < 1$ (2.39) implies that

$$\|\tilde{e}_{k+1}\| \leq \rho \|\tilde{e}_k\| \quad (2.40)$$

or

$$0 \leq \|\tilde{e}_k\| \leq \rho^k \|\tilde{e}_0\| \rightarrow 0 \text{ as } k \rightarrow \infty \quad (2.41)$$

In this case the algorithm (2.24) converges in the sense of (2.41). If we plug (2.35) in (2.37) we get

$$\text{tr}(F^T F) = [\gamma^2 \|P\|_F^2 - 2\gamma \text{tr}(P) + \text{tr}(I)] \quad (2.42)$$

For $\text{tr}(I) = n$ we get

$$\text{tr}(F^T F) = [\gamma^2 \|P\|_F^2 - 2\gamma \text{tr}(P) + n] \quad (2.43)$$

Equation (2.43) is quadratic in γ . There will be always an α for which (2.43) is < 1 except when

$$(\text{tr}(P))^2 \leq (n-1) \|P\|_F^2 \quad (2.44)$$

In this case ρ is ≥ 1 for all γ and the algorithm (2.24) diverges.

2.3 Remarks and Discussion

At the end of this subsection two points need to be mentioned: The first point is that as in the case of C^0 -type betterment it can be shown (using the L2-representation theorem) that if the transfer function of the causal system under test is proper, the algorithm (2.24) may diverge. The second point is that in this section the Forbenious norm was chosen mainly because it is relatively easy and inexpensive (computer time) to numerically compute. In future work other matrix norms will have to be tested and their relations to the rate of convergence of the algorithm should be determined.

Example 2.1

The applicability of the result of theorem 5 is illustrated in this example. Consider the first order linear time invariant system described by the following transfer function:

$$F(s) = 100 \frac{s + .89}{s + 0.9}$$

For easy checking with analytic result let the desired output $y_d(t)$ be:

$$y_d(t) = 3.46410t - 0.73205$$

which can also be written as in (2.6)

$$\tilde{y}_d(t) = c_1 \phi_1(t) + c_2 \phi_2(t)$$

where

$$c_1 = 1$$

$$c_2 = 1$$

and

$$\phi_1(t) = 1$$

$$\phi_2(t) = 2\sqrt{3}t - \sqrt{3}$$

are two Legendre functions. In this case $N = 2$ and

$$P = \begin{bmatrix} 99.62 & -.19 \\ .19 & 99.94 \end{bmatrix}$$

the condition (2.27) is satisfied and the algorithm (2.24) converges in two iterations as it is shown in figure(2.1)

3.0 Comparison of Learning Algorithms

In this part a comparison between the 2 methods introduced above is presented. Based on the decomposition technique and the analogy with the C^1 -type betterment an extension and generalization of the method of section 2 are presented.

3.1 An Extended Algorithm

Let us now use the decomposition technique on the C^1 -type betterment. Again using the same notation as in part 2 we will have :

$$\tilde{y}_d(t) = \underline{d}^T \Phi \quad (3.1)$$

$$\tilde{y}_k(t) = \underline{\alpha}^T(k) \Phi \quad (3.2)$$

$$\tilde{u}_k(t) = \underline{\beta}^T(k) \Phi \quad (3.3)$$

From the C^1 -type betterment formula we get

$$\tilde{u}_{k+1}(t) = \tilde{u}_k(t) + \delta \frac{d}{dt} \{ \tilde{y}_d(t) - \tilde{y}_k(t) \} \quad (3.4)$$

or

$$\underline{\beta}^T(k+1) \Phi = \underline{\beta}^T(k) \Phi + \delta (\underline{d}^T - \underline{\alpha}^T(k)) \Phi \quad (3.5)$$

Let

$$\begin{aligned} \dot{\phi}_1(t) &= \sum_{i=1}^n a_{1i} \phi_i(t) \\ &\vdots \\ \dot{\phi}_n(t) &= \sum_{i=1}^n a_{ni} \phi_i(t) \end{aligned} \quad (3.6)$$

or in vector form

$$\begin{bmatrix} \dot{\phi}_1(t) \\ \vdots \\ \dot{\phi}_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} \phi_1(t) \\ \vdots \\ \phi_n(t) \end{bmatrix} \quad (3.7)$$

therefore

$$\dot{\Phi} = A \Phi \quad (3.8)$$

Using (3.8) in equation (3.5) we get

$$\underline{\beta}^T(k+1) = \underline{\beta}^T(k) + \delta (\underline{d}^T - \underline{\alpha}^T(k)) A \quad (3.9)$$

Let P be the matrix of the operator L in the basis $[\phi_i(t)]$ i.e.

$$L(\Phi) = P\Phi$$

therefore

$$\underline{\alpha}(k) = P^T \underline{\beta}(k) \quad (3.10)$$

Now we have

$$\begin{aligned} \underline{\beta}^T(k+1) &= \underline{\beta}^T(k)\Phi + \delta(\underline{d}^T - \underline{\beta}^T(k)P)A\Phi \\ &= (\underline{\beta}^T(k)[I - \delta PA] + \delta \underline{d}^T A)\Phi \end{aligned} \quad (3.11)$$

$$\underline{\beta}(k+1) = [I - \delta A^T P^T] \underline{\beta}(k) + \delta A^T \underline{d} \quad (3.12)$$

If we let

$$PA = R \quad (3.13)$$

$$A^T \underline{d} = \underline{V} \quad (3.14)$$

then equation (3.12) becomes

$$\underline{\beta}(k+1) = [I - \delta R^T] \underline{\beta}(k) + \delta \underline{V} \quad (3.15)$$

Equation (3.15) and (2.24) are very similar therefore we can conclude that the C^0 -type betterment and C^1 -type betterment are just two different ways of writing the limiting case of algorithm (2.24). On the other hand, one should notice that equation (3.12) can be also arranged in the following way. Let

$$M = \delta A^T \quad (3.16)$$

then using (3.16) we can write (3.12) as

$$\underline{\beta}(k+1) = (I - MP^T) \underline{\beta}(k) + M \underline{d} \quad (3.17)$$

This important result is generalized and discussed in the next subsection.

3.2 Generalization Of The Algorithm

The method of section 2 and the extended algorithm of subsection 3.1 (equation 3.17)) can be combined in the following way. Assume (like in the case of mixed type betterment) that the ϵ -approximation of the $k+1$ input $u_{k+1}(t)$ is given by the following iteration:

$$\tilde{u}_{k+1}(t) = \tilde{u}_k(t) + \gamma(\tilde{y}_d(t) - \tilde{y}_k(t)) + \delta \frac{d}{dt} \{ \tilde{y}_d(t) - \tilde{y}_k(t) \} \quad (3.18)$$

then

$$\underline{\beta}^T(k+1) = \underline{\beta}^T(k)\Phi + \gamma(\underline{d}^T - \underline{\alpha}^T(k))\Phi + \delta(\underline{d}^T - \underline{\alpha}^T(k))\dot{\Phi} \quad (3.19)$$

let

$$L(\Phi) = P\Phi \quad (3.20)$$

therefore

$$\underline{\alpha}(k) = P^T \underline{\beta}(k)$$

which gives

$$\underline{\beta}^T(k+1) = \underline{\beta}^T(k)\Phi + \gamma(\underline{d}^T - \underline{\beta}^T(k)P)\Phi + \delta(\underline{d}^T - \underline{\beta}^T(k)P)A\Phi \quad (3.21)$$

$$\underline{\beta}^T(k+1) = \{\underline{\beta}^T(k)[I - \gamma P - \delta P A] + \underline{d}^T[\gamma I + \delta A]\}\Phi \quad (3.22)$$

which implies that

$$\underline{\beta}(k+1) = [I - \gamma P^T - \delta A^T P^T]\underline{\beta}(k) + [\gamma I + \delta A^T]\underline{d} \quad (3.23)$$

Let

$$Q = \gamma I + \delta A^T \quad (3.24)$$

then we can arrange (3.23) as

$$\underline{\beta}(k+1) = [I - QP^T]\underline{\beta}(k) + Q\underline{d} \quad (3.25)$$

where Q is a constant $n \times n$ gain matrix. Equation (3.25) can be therefore considered as a generalization of equation (3.17) and equation (2.24)

The convergence of the above algorithm is given by the following theorem

Theorem 6. For the problem considered in theorem 5 and a given desired trajectory $y_d(t)$ defined over the time interval $[t_1, t_2]$, and for a given set of orthogonal functions defined by the vector valued function Φ if there exists a constant $N \times N$ matrix Q which satisfies the following condition

$$\|I - QP^T\|_F < 1$$

then

$$\|\bar{e}_k\| \rightarrow 0 \text{ as } k \rightarrow \infty$$

Proof:

In the previous subsection we defined

$$\bar{e}_{k+1}(t) = \underline{z}^T(k+1)\Phi \quad (3.26)$$

where

$$\begin{aligned}\underline{z}(k+1) &= \underline{d} - \underline{a}(k+1) \\ &= \underline{d} - P^T \underline{z}(k+1)\end{aligned}\quad (3.27)$$

using iteration formula (3.25) we get

$$\underline{z}(k+1) = \underline{z}(k) - P^T Q \underline{z}(k) \quad (3.28)$$

so

$$\underline{z}(k+1) = (I - P^T Q) \underline{z}(k) \quad (3.29)$$

or

$$\underline{z}(k+1) = F \underline{z}(k) \quad (3.30)$$

where F is a $n \times n$ matrix defined by

$$F = I - P^T Q \quad (3.31)$$

If we use the Frobenius norm for the matrix F and the L2-norm for the vectors $\underline{z}(k+1)$ and $\underline{z}(k)$ then the inequality (2.30) implies that

$$\|\underline{z}(k+1)\| \leq \|F\|_F \|\underline{z}(k)\| \quad (3.32)$$

Let

$$\|F\|_F = \rho \geq 0$$

then by choosing Q such that

$$0 \leq \rho < 1$$

we have

$$\|\tilde{e}_k\| \rightarrow 0 \text{ as } k \rightarrow \infty$$

and the convergence is assured. From (3.25) it is clear that if P^{-1} exists then the best choice will be $Q = P^{-T}$. In this case we have $\underline{z}(k) = P^{-T} \underline{d}$ for all $k > 0$ and the algorithm converges in one step.

Example 3.1

Consider the following system:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

where

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [1 \ 0]$$

and the desired output is

$$y_d(t) = 12t^2(1 - t)$$

for t in $[0, 1]$. For this system the condition (2.27) of theorem 5 is not satisfied, however by choosing a set of 4 Legendre functions with a matrix Q

$$Q = \begin{bmatrix} -100.58 & 177.64 & -166.63 & 190.80 \\ -176.70 & 280.0 & -272.10 & 302.15 \\ -165.80 & 270.66 & -304.80 & 311.0 \\ -190.70 & 299.10 & -308.60 & 230.50 \end{bmatrix}$$

for which the condition of theorem 6 is satisfied, $\tilde{y}_k(t)$ converges to $\tilde{y}_d(t)$ in 4 iterations as it is shown in figure (3.1). We notice also that since in this example $CB = 0$, the conditions for the convergence of the C^1 -type betterment as has been shown in [3], [4] are not satisfied. Figure (3.2) shows the divergence of $y_k(t)$ from the desired trajectory $y_d(t)$ as the number of iteration k gets larger. However, for a system with the same matrices A and B as the system considered in this example but with matrix C equal to :

$$C = [0 \ 1]$$

the C^1 -type betterment converges as it is shown in figure (1.1).

3.4 Remarks And Discussion

In this part an iterative method which can be considered as the generalized version of the algorithm (2.24) was developed. This method is also based on the assumption that the system should be linear time invariant and continuous and that the desired output $y_d(t)$ should be given on the entire interval $[0, T]$. The Convergence of the algorithm, then depends only on the possibility of finding a matrix Q such that the Frobenius norm of the matrix F is less than unity. As far as the numerical considerations are concerned the algorithm developed in this part is easy to implement. Except for the decomposition of the desired output $y_d(t)$, there is no integration or derivation to be computed. Furthermore, this algorithm will converge in one step if the inverse of the matrix P can be numerically computed.

Bibliography

- [1] Tsytkin, Y.Z. "Adaptation and learning in automatic systems" Academic Press, New York 1971.

- [2] Goodwin, G.C. and Sin, K.S. "Adaptive filtering, prediction and control" Prentice-Hall, Inc., New Jersey 1984.
- [3] Arimoto, S. e.al "Learning control theory for dynamical systems" 24 th conference on decision and control FLORIDA December 1985 pp 1375-1380.
- [4] Arimoto, S. e. al. "Application of learning method for dynamic control of robot manipulators." 24th conference on decision and control FLORIDA December 1985 pp 1381-1386.
- [5] Fu, K.S. "Learning control systems -Review and outlook" IEEE transactions on automatic control. April 1970 pp 210-221.
- [6] Harris, C.J., and Valenca, J.M.E. "The stability of input - output dynamical systems " Academic Press 1983.
- [7] Dennis, J.E., and Schnabel, R. "Numerical methods for unconstrained optimization and nonlinear equations" Prentice-Hall, Inc. New Jersey 1983.
- [8] Robert, A.G., and Roberts, R.A "Signals and linear systems" John Wiley and sons , Inc. New York 1973.
- [9] Nering, E.D. "Linear algebra and matrix theory" John Wiley and sons , Inc. New York 1963.
- [10] Ljung, L., and Soderstrom, T. "Theory and practice of recursive identification" The MIT Press, Cambridge, Massachusetts 1983.
- [11] Mendel, J. "Discrete techniques of parameter estimation : the equation error formulation " Marcel Dekker, Inc. New York 1973.
- [12] Arimoto, S. e. al "Betterment operation of dynamical systems by learning a new control theory of servomechanisms or mechatronics" Proc. 23 rd IEEE LAS VEGAS Dec 1984.
- [13] Kailath, T. "Linear systems" Prentice - Hall, Inc. New Jersey 1980.
- [14] Aiserman, M.A. e.al "Theoretical Foundations of the potential functions method in pattern recognition" automation and remote control Vol 25 N 9 Sep. 1964 .

Figures

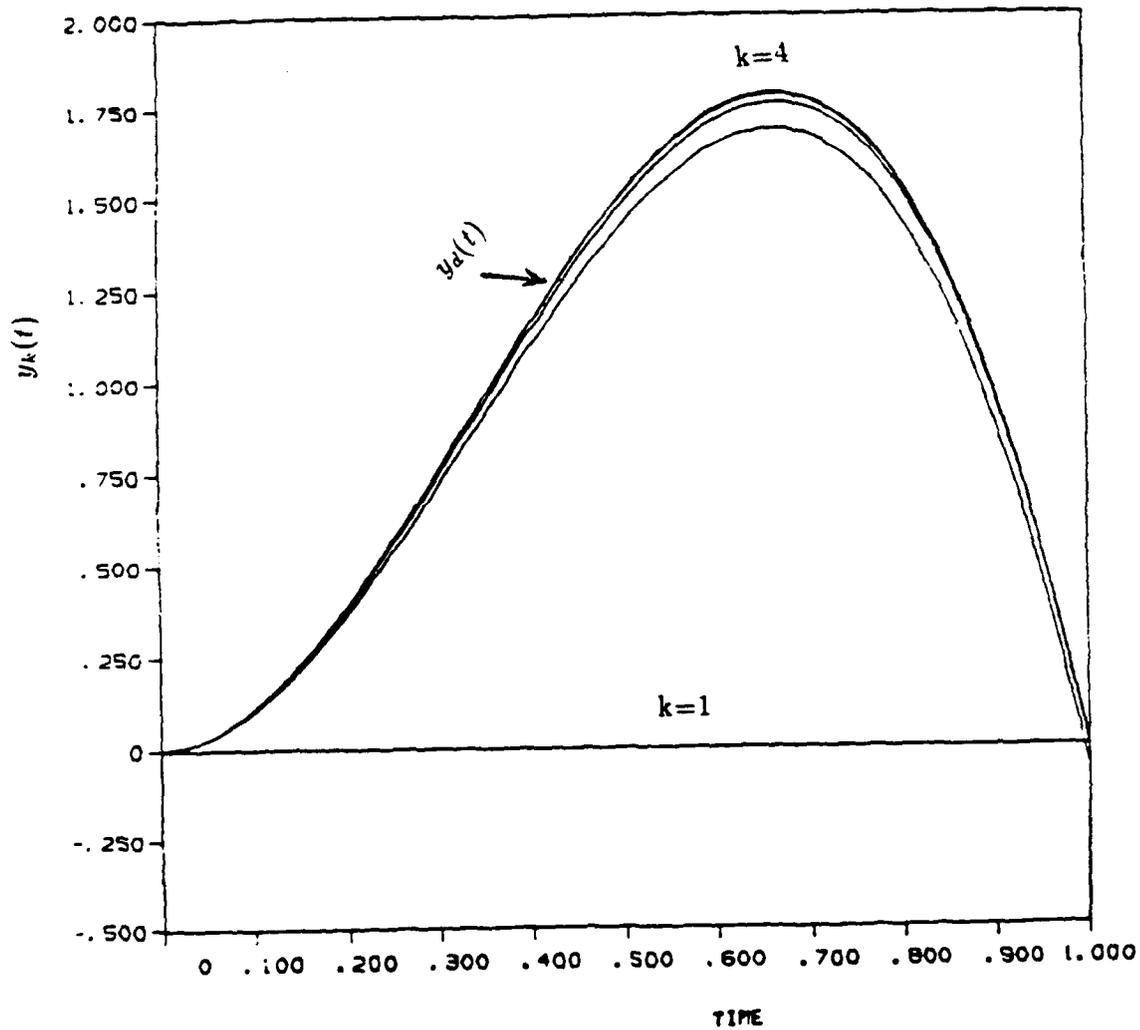


Figure 1.1

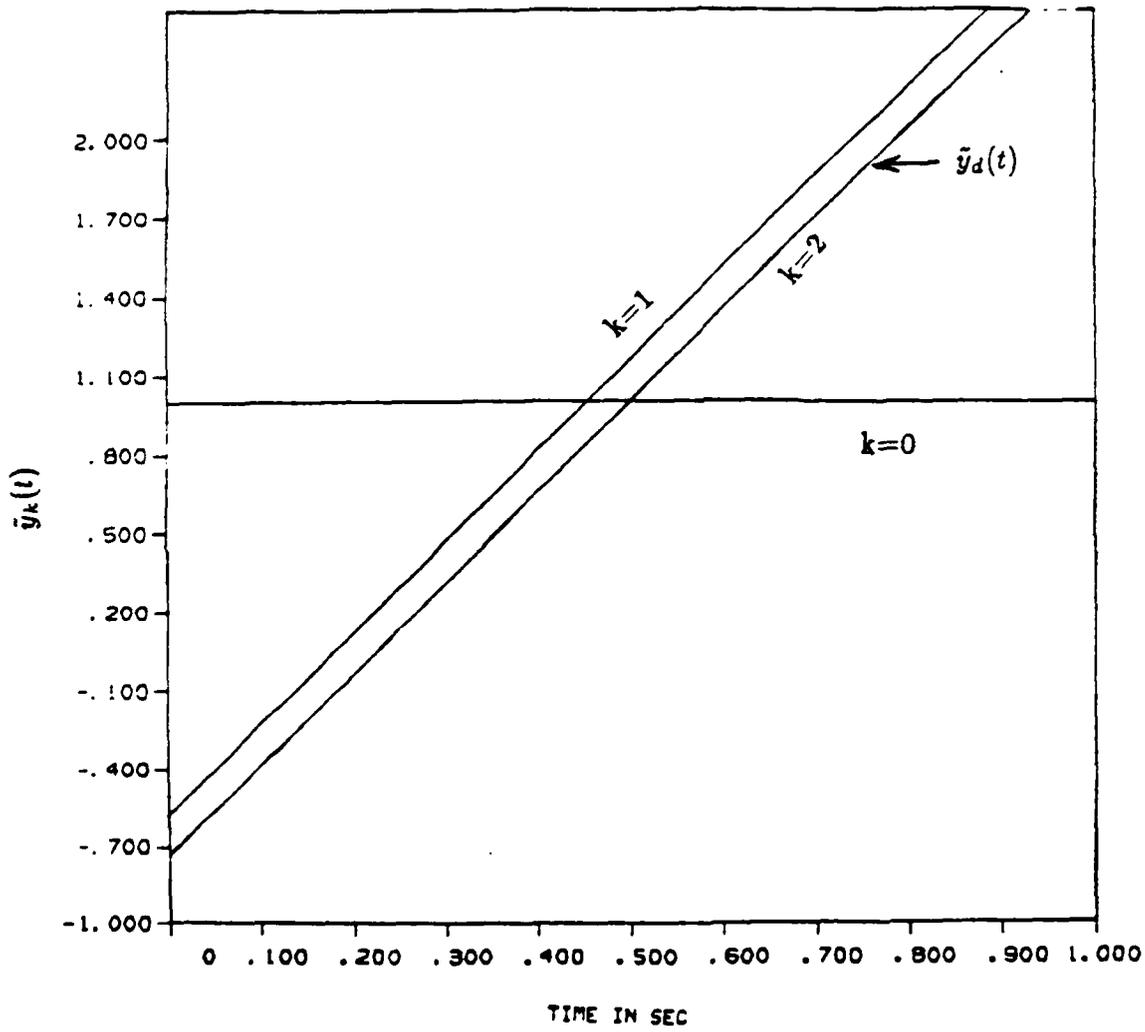


Figure 2.1

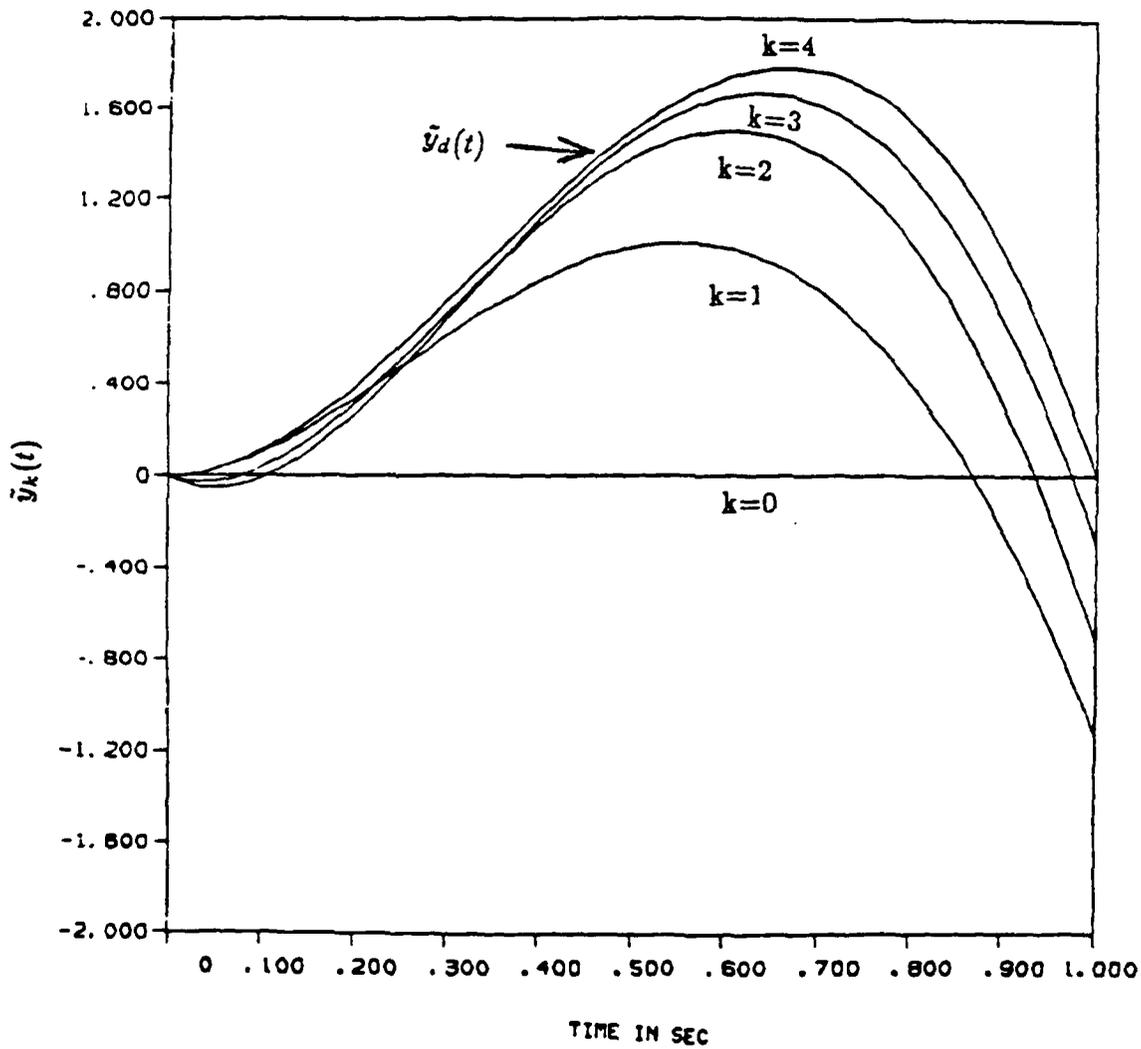


Figure 3.1

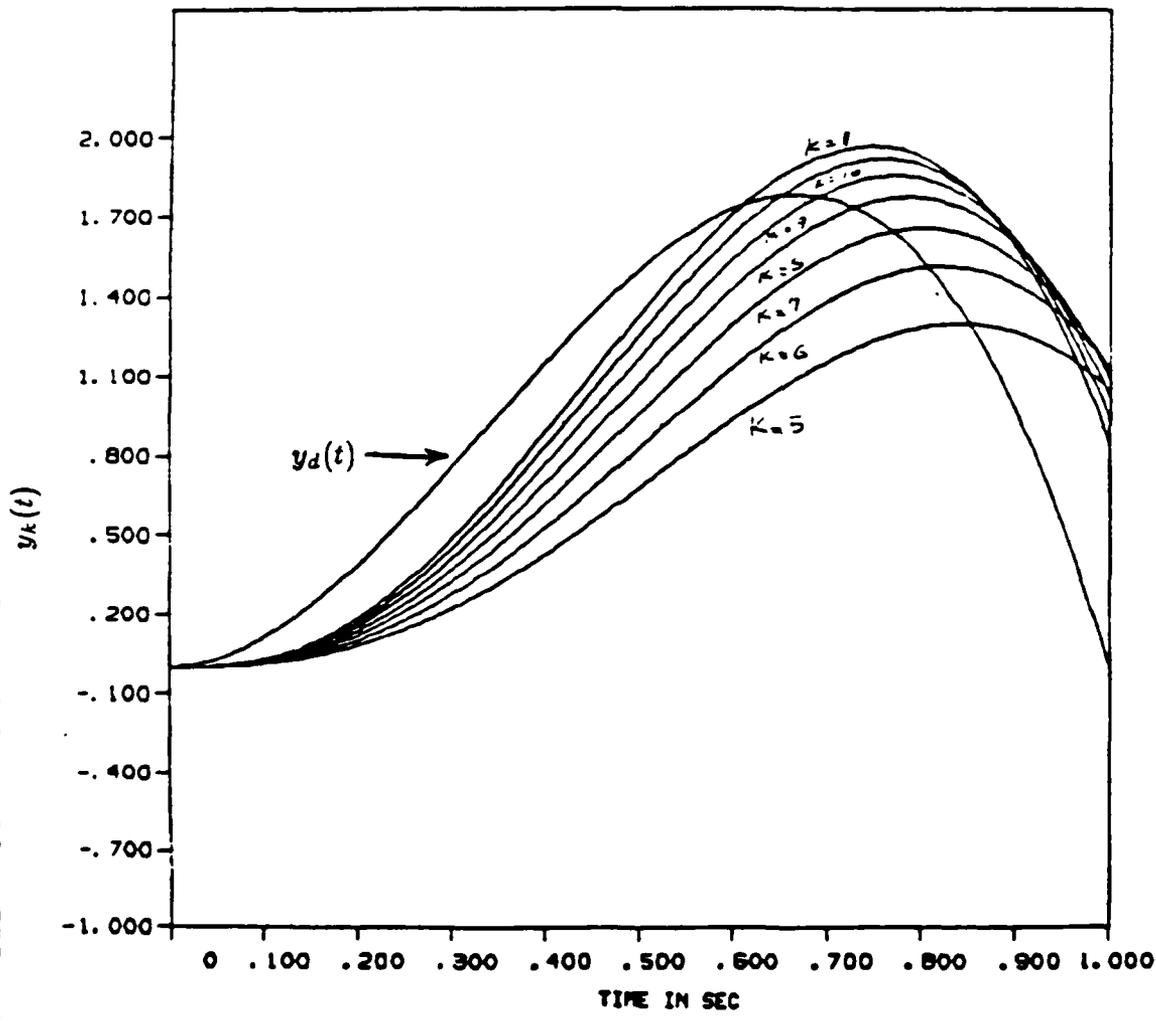


Figure 3.2

Part 4

A collection of papers on work which are in part supported by this ONR contract

Presented at the 18th Asilomar Conference on Circuits,
Systems and Computers, Nov. 4-7, 1984, Monterey, Calif.

On the Design of Time-Varying Controller for
Producing Desired System Responses*

Min-Yen Wu

Department of Electrical and Computer Engineering
University of Colorado, Boulder, CO 80309

Abstract:

It is well known that in linear time-invariant design, response speed and accuracy tend to be conflicting requirements that cannot be achieved simultaneously. For example, fast response always results in a large overshoot. Fast response without overshoot is desirable in many applications such as robot manipulator control. In this paper, it will be demonstrated that the use of time-varying controller can produce a response with short rise time, no overshoot and short settling time that are impossible to attain with conventional time-invariant design. Methods of designing the time-varying controller will be discussed. In particular, the method of dynamic pole assignment and the use of piecewise-constant time-varying feedback gain implementation will be emphasized.

left half complex plane, the time-varying controller may allow the overall system eigenvalues to be moving in the complex plane, even in the open right half complex plane for some intervals of time, provided that at the end the eigenvalues of the systems are placed at some desired steady-state pole locations in the open left half complex plane to insure the overall system stability. The idea of allowing the poles (the eigenvalues) to move around makes it possible to achieve a response with short rise time, no overshoot and short settling time, which are impossible to attain in time-invariant design. Several methods of designing the time-varying controller will be discussed. A new concept of employing dynamic pole assignment using piecewise constant time-varying feedback gain for the controller will be emphasized.

I. Introduction

In linear time-invariant system design, the performance requirements of fast response and good accuracy tend to conflict with each other. So compromises often have to be made. To ensure overall satisfactory performances, time-varying controller that adapts itself to yield optimal performances is needed. In this paper it will be demonstrated that the use of time-varying controller can produce a desired response with short rise time, no overshoot and short settling time, which are impossible to attain with conventional time-invariant design. It is shown that with time-varying controller, not only the overall performances are improved, but the system stability requirement is also greatly relaxed. In contrast to the linear time-invariant design which requires all eigenvalues (or the poles) of the system to be in the open

II. Controller Design Methods

The problem considered is as follows: Given a plant with transfer function $G(s)$ or it is defined by its state representation

$$\dot{\underline{x}}(t) = A \underline{x}(t) + Bu(t) \quad (1a)$$

$$y(t) = C \underline{x}(t) \quad (1b)$$

where $\underline{x}(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$. Also given is a desired response $y_d(t)$ to a given input. The objective is to design a controller so that $y_d(t)$ can be achieved.

If the requirements for $y_d(t)$ are such as the rise time, the overshoot and the settling time are not very stringent, then conventional lead-lag compensator [1] or the state feedback with constant gain matrix [2] will be adequate. However, if $y_d(t)$ is required to have very fast response without overshoot, for example, $y_d(t)$ is to approximate $y_d(t) = 1 - \exp(-t^2)$, then one has to resort to either time-varying or non-linear controller. Unfortunately, the stability of linear time varying system [3] or the non-linear systems [4] are very difficult to assure. In this paper, several controller design methods that will assure stability will be discussed. One particular method which uses piecewise constant time-varying state feedback will be emphasized.

*This research is supported in part by the Office of Naval Research under Contract N00014-K-0425.

(I) Optimal Control Tracking Problem Approach [5]

The optimal control theory can be used to design the controller so that the output of the system will track the desired response $y_d(t)$. The controller is to be designed according to some chosen performance index. In such design, the controller turns out to be a time-varying controller which requires the solution of a matrix Riccati operation. Hence the controller is difficult to implement. Furthermore, the choice of weighting matrices in the performance index for guaranteeing the desired response is not transparent.

(II) Multi-Segment Decomposition of Desired Response Approach

A time-invariant system may not produce the desired response over the entire interval of operation. However, by decomposing the desired response into several finite subintervals, say N , it is possible to approximate the desired response by a linear time-invariant system over each interval of interest. Although the system appears to be linear time-invariant over a given subinterval, it is time-varying and nonlinear over the entire course of operation. Under the multi-segment decomposition approach, two controller design methods are considered:

(a) Direct Control Law Generation Method

From the desired response $y_d(t)$ over the interval $t \in [t_i, t_{i+1}]$, $i=0, 1, 2, \dots, N-1$, one determines the corresponding desired state $x(t_i)$ and $x(t_{i+1})$. The control law $u(t)$ for $t \in [t_i, t_{i+1}]$ that will transfer $x(t_i)$ to $x(t_{i+1})$ is generated as follows:

$$u(t) = -B^T \exp[A_c(t-t_i)]Q^{-1}(t_i, t_{i+1}) \cdot \{x(t_{i+1}) - \exp[A_c(t-t_i)]x(t_i)\} \quad (2)$$

where $A_c(t_i, t_{i+1}) = A_c e^{A(t_i-t_{i+1})} - A_c e^{A(t-t_i)} - BB^T e^{A_c(t-t_i)}$

The controller is again a time-varying controller which is difficult to implement. One modified approach is to work with the discretized plant and to generate a sequence of discrete control laws. In this case, a multi-rate sampling may have to be employed, depending on the characteristics of the desired response which will call for non-equal length of segments in the decomposition of $y_d(t)$. This approach is not very straight forward.

(b) Piecewise Constant State Feedback Approach

In this approach, the desired response $y_d(t)$ is decomposed into segments to be approximated with a linear time-invariant system with desired pole locations. The desired response over the entire interval of operation is to be achieved through the dynamic management of pole locations. The time-varying controller will then consist of

a sequence of piecewise constant gain matrices which can be easily designed. The stability assurance and gain computations will be given in Section III. A design example that illustrates the procedure is given in Section IV.

III. Key Results for Piecewise Constant Time-Varying Controller Design

Theorem 1. (Stability Assurance)

Consider the system (1). Assume that (A, B) is completely controllable. Then there exists a set of constant matrices K_i , $i=0, 1, 2, \dots, N-1$ such that with the state feedback $u(t) = v(t) - K(t)x(t)$, where

$$K(t) = \begin{cases} K_i & t \in [t_i, t_{i+1}] \quad i=0, 1, 2, \dots, N-1. \\ K_{N-1} & t \geq t_N \end{cases} \quad (4)$$

the eigenvalues of $A(t) = A - BK(t)$ can be arbitrarily assigned over each interval $t \in [t_i, t_{i+1}]$. Furthermore, if all eigenvalues of $A_{N-1} = A - BK_{N-1}$ have negative real parts, then the system is stable.

Proof

The pole assignment part is well known because (A, B) is completely controllable. So only the stability part will be proved.

For any bounded input $v(t)$ we have for $t \in [t_i, t_{i+1}]$

$$x(t) = e^{A_c(t-t_i)} x(t_i) + \int_{t_i}^t e^{A_c(t-\tau)} B v(\tau) d\tau \quad (5)$$

where $A_c = A - BK_i$, $i=0, 1, 2, \dots, N-2$, and A_c is not required to be stable.

Since the linear system can not have finite escape time, hence $x(t) \neq \infty$ for all $0 < t < t_N$.

Now for $t > t_N$

$$x(t) = e^{A_{N-1}(t-t_N)} x(t_N) + \int_{t_N}^t e^{A_{N-1}(t-\tau)} B v(\tau) d\tau \quad (6)$$

Clearly $x(t) \neq \infty \forall t > t_N$ because $A(t) = A_{N-1}$ for $t > t_N$ and all eigenvalues of A_{N-1} have real parts. Hence $x(t) \neq \infty \forall t > 0$ and the system is stable.

Comment:

The only requirement for assuring the stability is that $A(t) = A_{N-1}$ for $t > t_N$ and A_{N-1} be stable. A_c 's for $i < N-1$ are not required to be stable. This flexibility permits some A_c to be unstable, particularly in the initial phase of operation, for fast response.

Lemma 1 (Bass-Guru formula)

Consider the system (1). Let $f(s)$ be the characteristic polynomial of A given by

$$d(s) = s^n + a_1 s^{n-1} + \dots + a_n \quad (7)$$

Let $\chi(s)$ be the desired characteristic polynomial for the system with the state feedback $u(t) = v(t) - Kx(t)$. Let $\chi(s)$ be given by

$$\alpha(s) = s^n + \alpha_1 s^{n-1} + \dots + \alpha_n \quad (8)$$

If the system (1) is completely controllable, then the state feedback gain K always exists and K can be computed by

$$K = (\underline{a}^T - \underline{\alpha}^T) S^{-1} Q^{-1} \quad (9)$$

where

$$\underline{a}^T = (a_1, a_2, \dots, a_n)$$

$$\underline{\alpha}^T = (\alpha_1, \alpha_2, \dots, \alpha_n)$$

S is a upper triangular Toeplitz matrix with first row $(1, a_1, a_2, \dots, a_{n-1})$ and

Q is the controllability matrix of (1) given by

$$Q = (B, AB, \dots, A^{n-1}B)$$

The proof can be found in [6], hence it is omitted.

Theorem 2 (Piecewise Constant Controller Design)

Consider the system (1). For a given input $v(t)$ let the desired response $y_d(t)$ be partitioned into N finite segments so that $y_d(t)$ for $t \in [t_i, t_{i+1})$ be characterized by a set of desired eigenvalues γ_{ij} , $i=0, 1, 2, \dots, N-1$, $j=1, 2, \dots, n$ and $\text{Re}(\gamma_{N-1,j}) < 0$ for all $j=1, 2, \dots, n$. Then there exists a piecewise constant time-varying state feedback $K(t)$, as defined in (4), that will produce a desired stable response. Furthermore, K_i in (4) can be computed from

$$K_i = (\underline{\alpha}_i^T - \underline{a}^T) S^{-1} Q^{-1} \quad i=0, 1, 2, \dots, N-1 \quad (10)$$

where $\underline{\alpha}_i^T = (\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in})$ is the row vector associated with the characteristic polynomial

$$\alpha_i(s) = s^n + \alpha_{i1} s^{n-1} + \dots + \alpha_{in} = \prod_{j=1}^n (s - \gamma_{ij}) \quad (11)$$

and \underline{a}^T , S , and Q are as defined in Lemma 1.

Proof

Theorem 2 follows from combinations of Theorem 1 and Lemma 1.

IV. Design Example

Consider a plant with transfer function $G(s) = 1/s^2$. Suppose the desired response $y_d(t)$ to a unit step input is as shown in Fig. 1.

Such a desired response can be described by

$$\ddot{y}(t) + 2\zeta(t)\dot{y}(t) + y(t) = 1 \quad (12)$$

with the damping ratio $\zeta(t)$ given by

$$\zeta(t) = \begin{cases} -1.0 & t \in [0, 0.7) \\ 0.5 & t \in [0.7, 1.3) \\ 4 & t \geq 1.3 \end{cases} \quad (13)$$

or equivalently by the desired eigenvalues

$$\begin{aligned} (\gamma_{01}, \gamma_{02}) &= (1, 1) \\ (\gamma_{11}, \gamma_{12}) &= (-0.5 \pm j 0.866) \\ (\gamma_{21}, \gamma_{22}) &= (-0.13, -7.87) \end{aligned} \quad (14)$$

It can easily be seen that for the implementation shown in Fig. 2 the piecewise constant time-varying controller gain $K(t)$ is given by

$$K(t) = \begin{cases} K_0 = (1, 2) & t \in [0, 0.7) \\ K_1 = (1, 1) & t \in [0.7, 1.3) \\ K_2 = (1, 8) & t \geq 1.3 \end{cases} \quad (15)$$

The pole movement in the complex plane is shown in Fig. 3

References:

- (1) Horowitz, I.M., Synthesis of Feedback Systems, Academic Press, New York, 1963.
- (2) Chen, C.T., Linear System: Theory and Design, Holt, Rinehart and Winston, 1984.
- (3) Wu, M.Y. "Stability of Linear Time-Varying Systems", Int'l Journal of Systems Science, Vol. 15, No. 2, Feb 1984, pp. 137-150.
- (4) Hsu, J.C. and A.U. Meyer, Modern Control Principles and Applications, McGraw-Hill, New York, 1978.
- (5) Kirk D.E., Optimal Control Theory, Prentice-Hall, Englewood Cliffs, 1970.
- (6) Kailath, T., Linear Systems, Prentice-Hall, Englewood Cliffs, 1980.

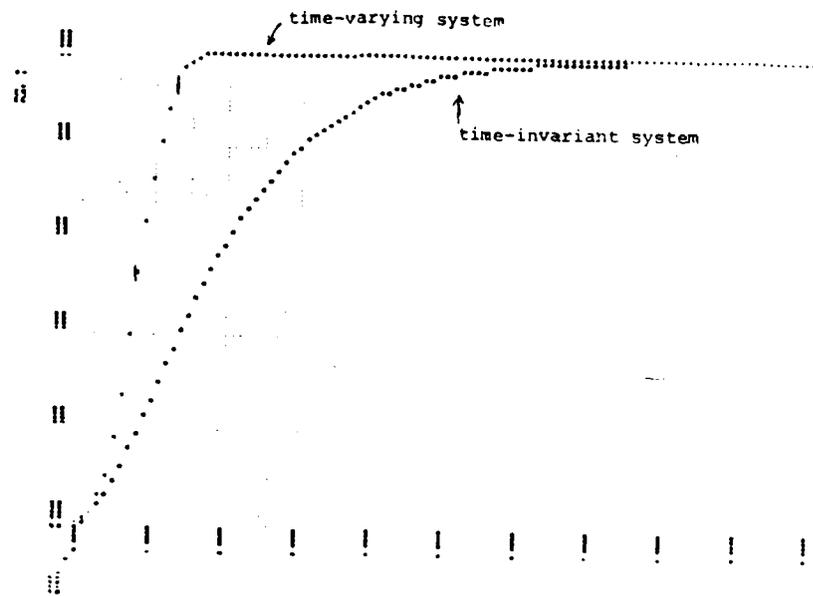


Fig. 1 Comparison of no-overshoot responses between time-invariant and time-varying systems

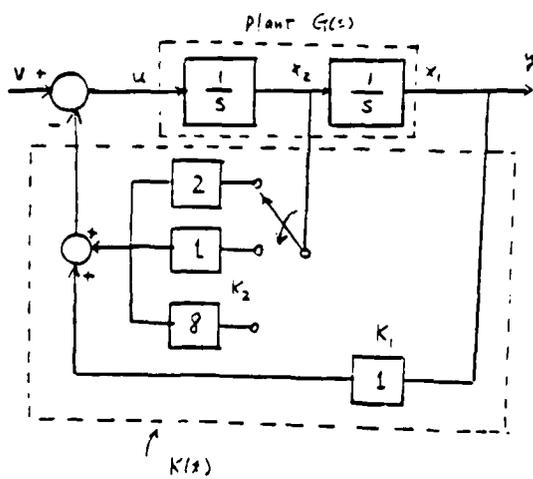


Fig. 2 System Implementation

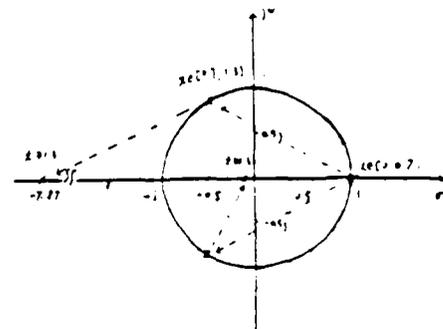


Fig. 3 Pole Movement

6th IASTED International Symposium on
Robotics and Automation 85
May 29-31, 1985 Santa Barbara, Calif.

Controller Design of a Multi-Joint Robot Manipulator

Min-ven Wu
N. M. Arbib

Department of Electrical and Computer Engineering
University of Colorado, Boulder, CO 80309

Abstract:

Feedback linearization theory in differential geometric control theory is used in conjunction with time-varying state feedback to design the time-varying controller for a multi-joint robot manipulator control. A 3-joint robot manipulator is used as an example to evaluate the controller performances. Simulation results show great promise of employing time-varying controller to achieve better dynamic control over that of time-invariant controller.

I. Introduction

Impelled by concerns about productivity, worker shortage, and hostile environments, interest in robotics has grown explosively in the last several years. Most industrial robots are basically computer controlled mechanical manipulators which can be programmed to perform the tasks, with minimum or no human intervention, of arc welding, paint spraying, assembly, foundry operation, etc. A manipulator consists of a series of links which are connected at joints. Typically they have three to six joints (three to six degrees of freedom) with a gripper or end effector. The joint can be either a revolute joint for rotational motion or a prismatic joint for translational motion. Each joint is driven by an actuator which is commanded by the controller.

The motion of the robot manipulator is desired to be fast, smooth, and accurate. The open-loop control method is not satisfactory because of variable inertia, gravity loading and load disturbances. Therefore the dynamic control of the robot manipulator is often of closed-loop control. The design of controller for the servo system depends on the dynamical model of the system to be controlled. The dynamic equations of a robot manipulator can be obtained either by Lagrangian formulation or the Newton-Euler formulation [1-2]. These dynamic equations are highly nonlinear and strongly coupled differential equations. In the past, the controller design is mostly based on independent joint control [1-2], linearization and feed-forward compensation [3-4]. One best known example of such approaches is the 'computed torque' technique [3] or the 'inverse problem' technique [5]. These techniques require either the manipulator links being weakly coupled, or each link being controlled one at a time and precise knowledge of robot manipulator dynamics be known. Such techniques become less effective for high speed and accurate control. It also often requires excessively large control torques or forces from the actuators.

Recent advances in differential geometric control theory [6-9] provide necessary and sufficient conditions for transforming a given nonlinear system around a given point into a local feedback-equivalent linear system. These results are found to be particularly useful to robotics applications because the Kronecker indices for such systems are mostly less than or equal to 2. With proper coordinate transformation and feedback transformation, the highly nonlinear and coupled robotic equations can be

transformed into a set of decoupled second-order linear dynamical equations. Consequently, various controller design techniques for linear systems become applicable. The use of feedback linearization approach to design robot controller are reported to give very good simulation results [7, 8, 10].

In this paper, we shall first show how the feedback linearization technique can be used to transform highly nonlinear and coupled robotic equations into a set of decoupled second-order linear system equations. Then controller design based on time-varying state feedback will be discussed. Finally, computer simulations for a 3-joint robot manipulator control are presented. The simulation results showed clearly the potential of a time-varying controller for achieving faster and accurate response with smaller control efforts than a conventional time-invariant controller.

II. Main Results

Definition 1. Kronecker Indices

The Kronecker indices for a matrix pair (A, B) are defined as follows: Let R_1 and L_1 be defined as:

$$R_1 = [B, AB, \dots, A^{l_1-1}B]$$

$$L_1 = \dim R_1 - \dim R_{1-1}$$

where $i = 1, 2, \dots$ are integers, $\dim R_1$ is the dimension of R_1 and $\dim R_0 = 0$. Then for an integer j , the Kronecker index K_j is defined to be the total number of l_i which are greater than or equal to j , i.e.,

$$K_j = \#\{l_i | l_i \geq j\}$$

Definition 2. Involutiveness

Let $C = \{X^1, X^2, \dots, X^k\}$ be a set of smooth vector fields in R^n with $X^1(p), \dots, X^k(p)$ being linearly independent for some point p . Then the set C is said to be involutive, if for any $X^i, X^j \in C$, there exists smooth, real valued functions $(\alpha_m^{ij}(x))_1^k$, such that the Lie product $[X^i, X^j](x) = \sum_{m=1}^k \alpha_m^{ij}(x) X^m(x)$ for x in the neighborhood of p .

Theorem 1. Multi-input Feedback Linearization Theorem

Consider the multi-input nonlinear system of the form

$$\dot{x} = X^0(x) + \sum_{i=1}^n u_i(x) X^i(x) \quad (1)$$

where $x \in R^n$, $X^0(x)$ and $X^i(x)$, $i = 1, 2, \dots, n$, are smooth vector fields in R^n and $u_i(x)$, $i = 1, 2, \dots, n$, are real valued smooth scalar functions. Let p be the point of

interest for linearization. The necessary and sufficient conditions for the system (1) with Kronecker indices K_1, K_2, \dots, K_n to be feedback equivalent to the linear time-invariant canonical form

$$\dot{z} = Az + Bu \quad (2)$$

where

$$A = \text{diag}(A_1, A_2, \dots, A_n) \text{ with } A_i = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \quad (3)$$

and $B = [b_1, \dots, b_n]^T$ with $b_i = [0, 0, \dots, 1, 0, \dots, 0]^T$ being an $n \times 1$ vector with 1 in the i th component and zeros elsewhere.

are

(i) There exist real $\{a_i\}_1^n$ such that

$$X^j(\underline{q}) + \sum_{i=1}^n a_i X^i(\underline{q}) = 0$$

(ii) $\dim C(\underline{q}) = n$ (local controllability test), where C is

$$C = (X^1, (\text{ad}^0 X^1), \dots, (\text{ad}^{K_1-1} X^1), X^2, (\text{ad}^0 X^2), \dots, (\text{ad}^{K_2-2} X^2), \dots, X^n, (\text{ad}^0 X^n), \dots, (\text{ad}^{K_n-1} X^n))$$

with $(\text{ad}^j X, Y) = Y, (\text{ad}^1 X, Y) = [X, Y], \dots, (\text{ad}^k X, Y) = [X, (\text{ad}^{k-1} X, Y)]$, being the Lie product notations $[X, Y] = X(Y) - Y(X)$ and X_i, Y_i are the Jacobian of X and Y with respect to \underline{x} .

(iii) For each $i=1, 2, \dots, n$, the set C_i defined as

$$C_i = (X^1, (\text{ad}^0 X^1), \dots, (\text{ad}^{K_i-2} X^1), X^2, (\text{ad}^0 X^2), \dots, (\text{ad}^{K_i-2} X^2), \dots, X_n, (\text{ad}^0 X^n), \dots, (\text{ad}^{K_i-2} X^n))$$

is involutive.

The proof for theorem 1 can be found in [6, 9].

Theorem 2. (Feedback Linearization of Robotic Equation)

Consider the dynamic equation of a n -joint robot manipulator, given by

$$D(\underline{q})\ddot{\underline{q}} + H(\underline{q}, \dot{\underline{q}}) + G(\underline{q}) = \underline{\tau} \quad (3)$$

where $\underline{q} = (q_1, q_2, \dots, q_n)^T$ is the joint position vector.

$D(\underline{q})$ is an $n \times n$ matrix representing effective and coupled inertia, $H(\underline{q}, \dot{\underline{q}})$ is an $n \times 1$ vector representing the Coriolis and Centrifugal terms and $G(\underline{q})$ is an $n \times 1$ vector representing the gravity loading terms. Let $q_i = x_{2i-1}$ and $\dot{q}_i = x_{2i}$, $i=1, 2, \dots, n$ be chosen as state variables and rewrite (3) as in (1), where $X^0(\underline{x}) = [x_2, f_1(\underline{x}), x_4, f_2(\underline{x}), \dots, x_{2n}, f_n(\underline{x})]^T$ and $X^1(\underline{x}) = [0, 0, \dots, 1, 0, \dots, 0]^T$, $i=1, 2, \dots, n$, is an $2n \times 1$ vector with 1 in the $2i$ element and zeros elsewhere, $u_i(\underline{x})$ is the control which contains a nonlinear combination of some torques or forces τ_i and $f_i(\underline{x})$ is a smooth real function of \underline{x} for all i .

Let \underline{q} be the equilibrium point of (1) for $u_i(\underline{x})=0$ for all i . Then the coordinate transformation $\underline{z}=\underline{x}$ and the feedback transformation $u_i(\underline{x}) = f_i(\underline{x}) + u_i$, $i=1, 2, \dots, n$ with u_i being a new free control, will transform (1) into a feedback equivalent linear time-invariant canonical form (2) where

$$A_i = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \forall i = 1, 2, \dots, n \quad (4)$$

and

$$b_i = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$$

is an $2n \times 1$ vector with 1 in the $2i$ element and zeros elsewhere.

Proof of Theorem 2:

First, we show that the Kronecker indices for system (1) corresponding to (3) are equal to or less than 2.

Since $R = X^0(\underline{q})$ has $K_i(\underline{q}) = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$ with 21 element being 1 and zeros elsewhere, clearly $L_i = 2, \forall i=1, \dots, n$. Now $R_i = X^0(\underline{q}), (\text{ad}^0 X^0)(\underline{q}), (\text{ad}^1 X^0)(\underline{q})$

So $R_i =$

$$R_i = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 \\ 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 & \frac{\partial f_i}{\partial x_{2i}}(\underline{q}) & 0 \end{pmatrix}$$

Hence $\dim R_i = 2n$. It follows that $L_i = n, \forall i=1, 2, \dots, n$ and $L_i = 0$ for $i > n$. Therefore the Kronecker indices are $K_i = 2, \forall i=1, 2, \dots, n$, and $K_i = 0$ for $i > n$. When $K_i \leq 2, \forall i$, the condition (iii) in theorem 1 is implied by condition (ii). So to establish feedback equivalence we need to check only conditions (i) and (ii) in theorem 1. Since $X^0(\underline{q})=0$, condition (i) is satisfied with $a_i=0, \forall i$. Since $\dim C(\underline{q}) = \dim R_i = 2n$, the system (1) is locally controllable at \underline{q} . So condition (ii) of theorem 1 is satisfied. In fact, all three conditions of theorem 1 are satisfied. Therefore system (1) can be made to be feedback equivalent to (2). More explicitly, it can be shown that the coordinate transformation $\underline{z}=\underline{x}$ and the feedback transformation $u_i(\underline{x}) = f_i(\underline{x}) + u_i$ will transform (1) into (2) with A_i and b_i as given in (4).

Comments:

1. Theorem 2 shows that with feedback linearization, the highly nonlinear and strongly coupled robotic equations can be transformed into a set of n decoupled second-order linear time-invariant canonical form with double poles at the origin.

2. The smoothness requirement on vector fields in (1) prevents the application of feedback linearization to system with violent nonlinearities such as stiction and coulomb friction which are always present in actuators at robot joints. Such effects can be removed from the dynamic model and be compensated later by feedforward compensation.

III. Controller Design

The objective of the robot manipulator control is to move the manipulator from a given joint position $x_1, x_2, \dots, x_{2n-1}$ to the desired joint position $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{2n-1}$.

Equivalently, via the coordinate transform, this is the problem of moving from the states $y_1, y_2, \dots, y_{2n-1}$ to desired states $\bar{y}_1, \bar{y}_2, \dots, \bar{y}_{2n-1}$. It is desirable to go

to the desired states in the shortest time possible without causing any overshoot or producing steady-state error in the response. It is also important to achieve such objectives with as little control efforts (the actuator torques or forces) as possible. Of course the simplicity in implementing the controller has to always be kept in mind. After the nonlinear robotic equation is converted into its feedback equivalent linear canonical form, various controller design techniques for linear systems become applicable. The conventional constant state feedback design tends to give slower responses or to require larger control efforts than those desired. The optimal error correcting controller [3] requires the extensive computations of the matrix Riccati equation. In this paper, we consider the use of time-varying state feedback controller of the following form:

$$u_i = F_i(x) + K_{1i}(t)y_{1i} + K_{2i}(t)y_{2i} \quad (5)$$

where $K_{1i}(t)$ and $K_{2i}(t)$ are respectively the time-varying position and velocity feedback gains, y_{1i} is the reference input and y_{2i} is the new free control in the linear canonical form.

Using the relations in coordinate transformation and feedback transformation, the control function $u_i(x)$ in (5) is given by

$$u_i(x) = F_i(x) + K_{1i}(t)(x_{2i+1} - x_{2i+1}^d) + K_{2i}(t)(x_{2i+1} - x_{2i+1}^d) \quad (6)$$

where $F_i(x)$ is as given in theorem 2, x_{2i+1}^d and \dot{x}_{2i+1}^d ($i=1, 2, \dots, n$) are the given and desired joint positions.

The actuator control torques or forces $u_i(t)$, $i=1, 2, \dots, n$ are obtained by solving for u_i from the sets of equations $u_i(x)$, $i=1, 2, \dots, n$.

In practice, all actuators have their maximum limits on their control torques or forces. So in designing the controller, the constraint $|u_i(t)| \leq \bar{u}_i$ for all t is imposed. The time-varying feedback gain $K_{1i}(t)$ and $K_{2i}(t)$ will be so chosen to maximize the response speed and the accuracy without causing the control torques or forces to exceed its limits. To maintain critically damped response for all t , it is required that $K_{2i}(t) = 2\sqrt{K_{1i}(t)}$.

The procedures for designing the time-varying controller are summarized below:

1. Initialize $K_{1i}(t)$ and input F_i .
2. Compute $K_{2i}(t) = 2\sqrt{K_{1i}(t)}$.
3. Read x_{2i+1}^d , \dot{x}_{2i+1}^d and x_{2i+1} .
4. Compute $u_i(x)$ from (6) and solve for u_i .
5. Check to see if $|u_i| \leq \bar{u}_i$.
6. Based on step 5, update $K_{1i}(t)$ and $K_{2i}(t)$.
7. The control cycle is done when \dot{x}_{2i+1}^d is reached for all $i=1, 2, \dots, n$.

Discussions on updating $K_{1i}(t)$ and $K_{2i}(t)$.

The update of time-varying gains $K_{1i}(t)$ and $K_{2i}(t)$ can be done by employing the dynamic pole management technique [11] or other updating schemes. The updating scheme we used for simulations to be discussed in the next section is to double $K_{1i}(t)$ when $|u_i| < \bar{u}_i$ and to reduce $K_{1i}(t)$ by half when $|u_i| > \bar{u}_i$. Such a scheme is employed primarily for its ease in implementation. Our main objective here is to demonstrate the potential and the capability of time-varying controller to yield more satisfactory response than that of a time-invariant controller.

10. Example and Simulation Results

A 3-joint robot manipulator shown in Fig. 1 is used as an example to evaluate the performances of controller design concepts described in this paper.

The 3-joint manipulator of Fig. 1 consists of one revolute joint and two prismatic joints. The rotary joint rotates in the xy plane (joint variable θ_1). One of the prismatic joints allows the hand to extend in the xy plane (joint variable y_1) while the other lets the hand translate along the z -axis (joint variable z_1). The robotic equations and controller design based on feedback linearization and time-varying feedback gains are omitted here due to lack of space. Their details can be found in [10]. For the following parameters: $M_1=4\text{kg}$, $M_2=10\text{kg}$, $l_1=1\text{m}$, $l_2=0.5\text{m}$ (or N) θ_1^d , z_1^d , y_1^d for torque or force limits, we simulated the joint motions from $(y_1=1\text{m}, z_1=0\text{rad}, \theta_1=0)$ to $(y_1=0.5\text{m}, z_1=0.5\text{rad}, \theta_1=0.2\text{rad})$. The results are shown in Table 1 and Figures 2, 3, and 4. These results when compared with those using computed torque technique or feedback linearization with constant state feedback show much improvements in convergence time, steady-state accuracy and control efforts. (See Tables 2 and 3.)

Acknowledgement

This research was supported in part by the Office of Naval Research under Contract N00014-K-0425.

References

- [1] Paul, R. P., Robot Manipulators: Mathematics, Programming, and Control, The MIT Press, 1981.
- [2] Brady, M. J., M. Hollerbach, T. L. Johnson, T. Lozano-Perez and M. T. Mason, Robot Motion Planning and Control, The MIT Press, 1982.
- [3] Bejczy, A. K., "Robot Arm Dynamics and Control," Jet Propulsion Laboratory, Tech. Report 33-609, Feb. 1974.
- [4] Vukobratovic, M. and D. Stokich, Control of Manipulation Robots, Springer-Verlag, Berlin, 1982.
- [5] Paul, R. P., "Modeling, Trajectory Calculation and Servicing of a Computer Controlled Arm," Stanford Artificial Intelligence Laboratory, Stanford Univ., AIM 177, 1972.
- [6] Hunt, L. R., R. Lu and G. Meyer, "Design for Multic-

- [6] Hunt, L. R., R. Su and G. Meyer, "Design for Multi-Input Nonlinear Systems," Proc. the Conference on Differential Geometric Control Theory, MTU, June 29 - July 2, 1982, pp. 268 - 298.
- [7] Tarn, T. J., A. K. Bejczy, A. Isidori and Y. Chen, "Nonlinear Feedback in Robot Arm Control," Proc. 23rd IEEE Conference on Decision and Control, Las Vegas, NV, December 12 - 14, 1984, pp. 736 - 751.
- [8] Bejczy, A. K., T. J. Tarn and Y. L. Chen, "Robot Arm Control by Computer," Proc. 1985 IEEE Intl. Conference on Robotics and Automation, St. Louis, MO., March 25 - 28, 1985, pp. 360 - 370.
- [9] Hermes, H. Lecture Note Math 542, University of Colorado, Boulder.
- [10] Arbouz, W. M. Robot Controller Design, M.S. Thesis, University of Colorado, Boulder, May, 1985.
- [11] Wu, M. Y., "On the Design of Time-Varying Controller for Producing Desired System Responses," Proc. the 18th Asilomar Conference on Circuits, Systems and Computers, Nov. 4 - 7, 1984, Monterey, CA.

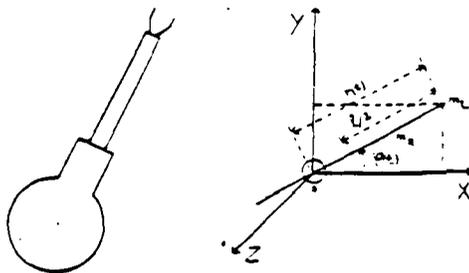


Figure 1 Three-joint manipulator

Table 1 Simulation of joint motion from (1,0,-1) to (0.5,-0.6,-0.2)

THREE JOINT ROBOT MANIPULATOR-FEEDBACK LINEARIZATION, TIME-VARYING FEEDBACK GAINS

TIME	θ	φ1	φ2
0	0.0000	0	0.0000
05000	08000	- 01840	- 07576
10000	04298	- 08348	- 00477
15000	08498	- 14279	- 11637
20000	01722	- 22629	- 08990
25000	12872	- 32374	- 07238
30000	05193	- 41140	- 04208
35000	07464	- 50443	- 02763
40000	03819	- 60402	- 01184
45000	00167	- 69872	- 00171
50000	00000	- 00000	- 00000

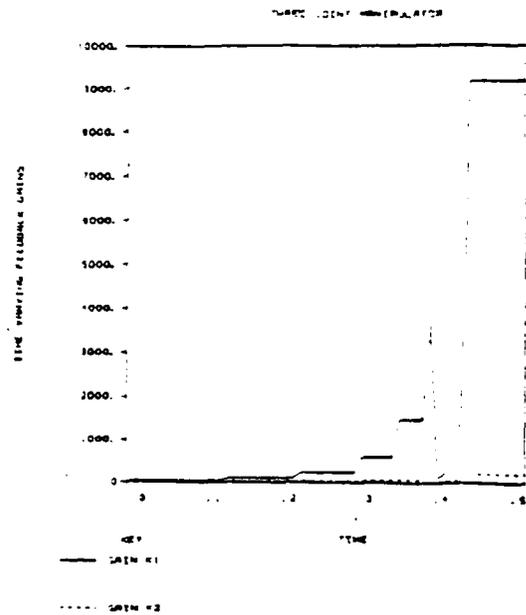


Figure 2 Feedback linearization. Time varying feedback gains K_1, K_2 .

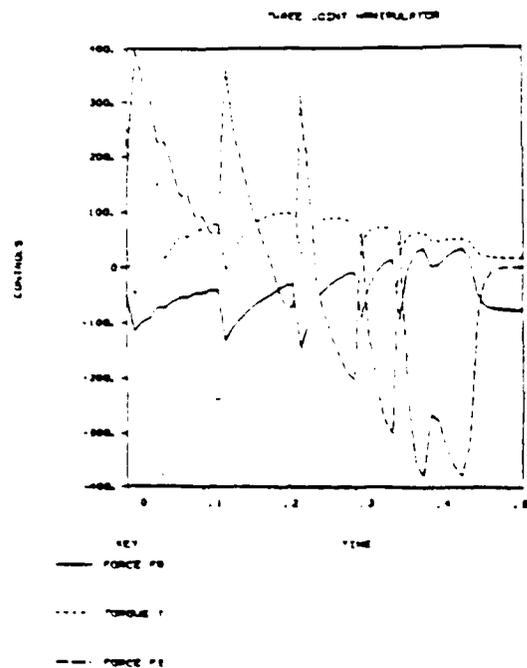


Figure 3 Feedback linearization with time varying feedback gains. Controls.

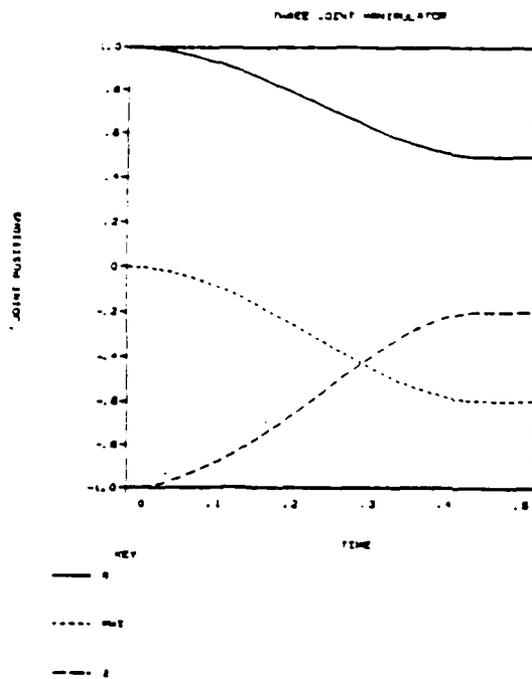


Figure 4. Feedback linearization with time varying feedback gains. Simulation of joint motions from (1,0,-1) to (0.5,-0.8,-0.2).

Table 3. Simulation of joint motions from (1,0,-1) to (0.5,-0.8,-0.2).

THREE JOINT ROBOT MANIPULATOR-FEEDBACK LINEARIZATION ON CONSTANT FEEDBACK GAINS

TIME	X	PHI	Z
0	1.0000	0	0.0000
10000	0.9480	-0.0424	-0.0257
20000	0.8973	-0.0872	-0.0528
30000	0.8487	-0.1345	-0.0812
40000	0.8020	-0.1834	-0.1108
50000	0.7573	-0.2330	-0.1413
60000	0.7148	-0.2833	-0.1725
70000	0.6743	-0.3344	-0.2042
80000	0.6358	-0.3862	-0.2362
90000	0.6000	-0.4388	-0.2692

Table 2. Simulation of joint motions from (1,0,-1) to (0.5,-0.8,-0.2) with $K_p = 1000$.

THREE JOINT ROBOT MANIPULATOR-COMPUTED TORQUE TECHNIQUE ON CONSTANT FEEDBACK GAINS

TIME	X	PHI	Z
0	1.0000	0	0.0000
10000	0.9479	-0.0401	-0.0239
20000	0.8970	-0.0821	-0.0502
30000	0.8478	-0.1262	-0.0762
40000	0.8010	-0.1722	-0.1022
50000	0.7570	-0.2192	-0.1282
60000	0.7150	-0.2672	-0.1542

Appendix: Three joint manipulator

The dynamic equations of the three joint manipulator can be written as:

$$\begin{aligned} \ddot{r} &= \frac{b_2}{b_1} \dot{r}^2 - \frac{3}{8} \frac{b_4}{b_1} \dot{\phi}^2 - \frac{b_3}{b_1} g \sin \phi + \frac{F_R}{b_1} \\ \ddot{\phi} &= -2b_2 \frac{r\dot{\phi}}{c_1} + \frac{3}{4} b_4 \frac{\dot{\phi}^2}{c_1} - b_3 g \frac{r \cos \phi}{c_1} - \frac{1}{2} b_4 \frac{\cos \phi}{c_1} - \frac{F}{c_1} \\ \ddot{z} &= \frac{F_z}{\sigma_3} \end{aligned} \quad (A1)$$

where

$$\begin{aligned} b_1 &= m_L + \frac{m_R}{4} \\ b_2 &= m_L + \frac{3}{4} m_R \\ b_3 &= m_L + m_R \\ b_4 &= m_R^2 \end{aligned} \quad (A2)$$

and

$$c_1 = b_2 r^2 + \frac{1}{4} b_4 (2-3r) . \quad (A3)$$

Let the state variables be:

$$\begin{aligned} x_1 &= r & x_3 &= \theta & x_5 &= z \\ x_2 &= \dot{r} & x_4 &= \dot{\theta} & x_6 &= \dot{z} \end{aligned} \quad (A4)$$

The state representation of (A1) is then,

$$\left\{ \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(x) + u_1(x) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(x) + u_2(x) \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= u_3(x) \end{aligned} \right. \quad (A5)$$

where

$$c_1(x) = b_2 x_1^2 + \frac{1}{4} b_4 (2-3x_1) \quad (A6)$$

$$f_1(x) = \frac{b_2}{b_1} x_1 x_2^2 - \frac{3}{8} \frac{b_4}{b_1} x_2^2 \quad (A7)$$

$$u_1(x) = -\frac{b_3}{b_1} g \sin x_3 + \frac{F_R}{b_1} \quad (A8)$$

$$f_2(x) = -2b_2 \frac{x_1 x_2 x_4}{c_1(x)} + \frac{3}{4} b_4 \frac{x_2 x_4}{c_1(x)} \quad (A9)$$

$$u_2(x) = -b_3 g \frac{x_1 \cos x_3}{c_1(x)} + \frac{1}{2} b_4 g \frac{\cos x_3}{c_1(x)} + \frac{T}{c_1(x)} \quad (A10)$$

$$u_3(x) = \frac{Fz}{b_3} \tag{A11}$$

System (A1) can be expressed as:

$$\dot{x} = X^0(x) + X^1(x)u_1(x) + X^2(x)u_2(x) + X^3(x)u_3(x) \tag{A12}$$

where

$$X^0 = \begin{bmatrix} x_2 \\ f_1(x) \\ x_4 \\ f_2(x) \\ x_6 \\ 0 \end{bmatrix}, \quad X^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad X^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad X^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$u_1(x), u_2(x), u_3(x)$ are the controls and $P = 0$ is the equilibrium point of (A5).

We check again the Kronecker indices.

$$k_1 = \dim R^1 = \dim \{X^1(P), X^2(P), X^3(P)\} = 3$$

let

$$R^2 = \{X^1(P), (adX^0, X^1)(P), X^2(P), (adX^0, X^2)(P), X^3(P), (adX^0, X^3)(P)\}$$

$$(adX^0, X^1)(P) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -2b_2 \frac{x_1 x_4}{c_1} + \frac{3}{4} b_4 \frac{x_4}{c_1} \\ 0 \\ 0 \end{bmatrix} (P) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(\text{ad}X^0, X^2)(P) = \begin{bmatrix} 0 \\ 2\left(\frac{b_2}{b_1}x_1 - \frac{3}{8}\frac{b_4}{b_1}\right)x_1^2 \\ 1 \\ -2b_2\frac{x_1x_2}{c_1} + \frac{3}{4}b_4\frac{x_2}{c_1} \\ 0 \\ 0 \end{bmatrix} (P) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(\text{ad}X^0, X^3)(P) = (0, 0, 0, 0, 1, 0)^T$$

Hence $\dim R^2 = 6$ and $\lambda_2 = \dim R^2 - \dim R^1 = 3$, $\lambda_i = 0$ $i \geq 3$.

The Kronecker indices are given by:

$$K_1 = K_2 = K_3 = 2, \quad K_i = 0 \quad i \geq 4$$

Thus conditions 1 and 3 of the feedback theorem are satisfied.

Also $C(P) = R^2$. Thus, $\dim C(P) = 6$, which makes condition 3 true.

System (A5) is then feedback equivalent to the following linear canonical form (A13):

$$\left\{ \begin{array}{l} \dot{y}_1 = y_2 \\ \dot{y}_2 = u_1 \\ \dot{y}_3 = y_4 \\ \dot{y}_4 = u_2 \\ \dot{y}_5 = y_6 \\ \dot{y}_6 = u_3 \end{array} \right. \quad (A13)$$

where u_1, u_2 and u_3 are new free controls. Again, the real valued functions h_1, h_2, h_3 are chosen as:

$$\begin{aligned} h_1(x) &= x_1 \\ h_2(x) &= x_3 \\ h_3(x) &= x_5 \end{aligned} \quad (A14)$$

which leads to an identity coordinate change. The feedback controls u_1, u_2 and u_3 are given by:

$$\begin{aligned} u_1(x) &= -f_1(x) + K_1(w_1 - x_1) - K_2 x_2 \\ u_2(x) &= -f_2(x) + K_3(w_3 - x_3) - K_4 x_4 \\ u_3(x) &= K_5(w_5 - x_5) - K_6 x_6 \end{aligned}$$

The explicit control forces F_R, F_z and torque T are obtained from (A8), (A11) and (A10)

$$F_R = b_1[-f_1(x) + K_1(w_1 - x_1) - K_2 x_2] + b_3 g \sin x_3 \quad (A15)$$

$$T = c_1(x)[-f_2(x) + K_3(w_3 - x_3) - K_4 x_4] + b_3 x_1 \cos x_3 - \frac{1}{2} b_4 g \cos x_3 \quad (A16)$$

$$F_z = b_3[K_5(w_5 - x_5) - K_6 x_6] \quad (A17)$$

These control forces F_R , F_z and control torque T , transform the nonlinear system (A5) (robot model) into the linear canonical system (A13).



*5/18/87
Miguel*

ROBOTICS

AND

AUTOMATION

**Santa Barbara, California, U.S.A.
May 27-29, 1987**

EDITOR: M.H. Hamza

**A Publication of
The International Association of Science
and Technology for Development - IASTED**

ISBN 0-88986-116-1

ACTA PRESS

ANAHEIM • CALGARY • ZURICH

Some New Thoughts on Control Design Strategies

Min Yen Wu and M. F. Chouikha
Department of Electrical and Computer Engineering
University of Colorado
Boulder, Colorado 80309

ABSTRACT

This paper describes some of our thoughts on future control design strategies. The controller design is viewed as a general decision making process. To control complex systems where precise dynamical models are not available, intelligent control should be used. An intelligent controller is a knowledge-based, rule-based, hierarchical controller with variable structure. It optimizes multiple objectives with constraints. It facilitates its decision making by control enrichment and observation enrichment. It has memory and is capable of learning from its experiences. Heuristic and qualitative rules are considered to be more important than quantitative and deductive methods in such decision making. As such, input-output (cause-effect) characterization of dynamical system is emphasized. Design of such intelligent controller calls for the possible integration of artificial intelligence and operational research with the control theory. Finally, it is suggested that integrated control design, which considers jointly the plant, actuators, sensors and controller at the outset of control design, should replace the current approach of designing a controller for a given plant.

I. Introduction

The majority of control design theories today require rather precise information on the parameters of the plant to be controlled. Most of the controller design is based on the system parameters such as poles, zeros or eigenvalues. In the case of linear systems, very often we begin with the system description of $\dot{x} = Ax + Bu$, $y = Cx + Du$ or its transfer function $G(s)$. The desired system response is characterized in terms of desired pole-zero pattern, desired eigenvalues, or optimization of performance index. Controller is then designed through the use of the pole-zero cancellation [1-2], loop shaping [2-3], state feedback [4-6] or performance index optimization [7-8]. In almost every case, the precise information of the plant parameters, such as poles and zeros of $G(s)$ or the matrices A , B and C , must be known in order to design the controller. In the case when plant parameters are not known, system identification [5] must be done first before the controller design can proceed. In the frequency-domain design, pole-zero information on the plant is needed for pole-zero cancellation and loop shaping. In the time-domain design, matrices A , B and C are needed for checking the controllability and observability and for computing the gain matrices for state feedback and state estimator. They are also needed in the solution of matrix Riccati equation in the optimal control design. In other words, explicit and precise information of system parameters is indispensable in most currently available controller design methods. In practice, many control systems, such as robotic systems in manufacturing, the discrete-event systems in production lines, the large structure control in space and the socio-economic systems, etc. often lack precise dynamical models due to complexity of the system, high degree of nonlinearity, large uncertainty, large parameter variations and time-varying parameters. It is becoming more apparent that conventional controller design theories are either inadequate or impractical for dealing with control of such systems. Therefore, new thinking, new approaches, and possibly new tools will be needed if we are going to control complex systems effectively. With the advent of learning theory in psychology, artificial intelligence in computer science, operational research in optimization and the generalization of control theory into a broader

decision-making theory, a new field of intelligent control has emerged [8-9]. This profound change in control concept undoubtedly will drastically change our future control design strategies. In this paper, we would like to present some of our thoughts on control design issues.

II. An Inspiration

Before we proceed to discuss various control strategies, let us first take a look at a well known control problem in our daily life, namely driving of a car. In car driving, the human is the controller. In fact, it is an intelligent controller. When we decide to go from point A to point B by way of car, we first find out when we have to get there and then try to drive there the best way we can. This may sound easy. However, it needs the best driving skill from an intelligent driver. When we say "the best way" it means somewhat differently from what is conventionally used in optimal control theory. To go from A to B in "a best way" may mean to take the shortest path to get there in minimum time, with minimum energy expenditure and to have a smooth and most enjoyable drive, etc. In other words, intelligent control often has to deal with fuzzy system requirements, to work with multiple objectives and to optimize performance indices which may not be optimal in strict mathematical sense. To choose the shortest path, the driver must know how many different paths are available and their respective lengths. To drive there in minimum time, he needs to go at maximum speed with constraints on speed limit and traffic and road conditions. To drive with minimum energy he needs good driving control of the car. So an intelligent controller should have a good task planning if it is going to have good control strategy. To have good task planning, it needs good knowledge base which comes from instructions or acquires through learning experiences. As to the actual control of the moving vehicle, a good control law generated by the controller (the driver) is needed. With the current control method, we would have to have a dynamical model with known parameters for the car we are driving, i.e. some differential equations characterizing the dynamics of the car on the road. With this parameter information, we will then figure out the gain matrix for the state feedback or the controller poles and zeros so that a satisfactory control law can be generated. As we know, when we drive we never bother to find out what are the dynamical equations, (the state equations or the transfer function) we are working with. If we had this information, we might be able to drive the car simply with the gas pedal control. If this were the case, then very few of us will be able to be a satisfactory driver (because the control strategies will be too complicated!). Of course, the car was not designed to be controlled by the gas pedal alone (even though the speed and direction of the car can be controlled by the gas pedal alone if we know the dynamical equations which accounts for the slope or curvature of the road). As we have seen, the single input control can be very ineffective in a complex control environment. To make control strategies simple and effective, we should expand our means of controlling the system. In the case of the car, we incorporated the brake and the steering wheel. The brake is to control the vehicle speed more effectively by providing fast deceleration, and for deceleration only. The steering wheel allows effective change of vehicle direction without the need of resorting to proper acceleration and deceleration which require complex computations involving the dynamical equations and the disturbances. The above observations suggest that effective control can be achieved more effectively by expanding the control capability of the system, namely through the control enrichment strategy. Now let us turn to the question of decision making in the controller. To

decide when, and how much to apply to the gas pedal, the brake or the steering wheel, the driver (human controller) just like in many conventional controllers, makes his decision based on his interpretation of the error signals (such as how close is the car from the side lines of the road, how close is the car in the front, how much is the speed, how curvy is the road, etc.). In the interpretation of error signals, the driver focuses his attention first on the polarity of the error and then on the quantitative aspects. In other words, the driver focuses primarily on qualitative aspects of the error signals to help him decide whether he should step on the gas pedal (to speed it up) or on the brake (to slow it down); whether to turn to the right or to turn to the left. The quantitative control such as how much to change the speed or the direction, though important, tends to be more gradual and is of secondary importance in nature.

The above observations suggest that although both the polarity and the magnitude of error signal should be taken into consideration in designing control strategies, clearly there should be a difference in their priority ranking. The error polarity is more important than the error magnitude. There seems to be very little compromise on error polarity in decision making. Nevertheless, there appears to exist a great deal of flexibility for reducing the error magnitude. It is also to be noted that generations of good control law requires good observations and feedback. Good driving requires good front, rear and side views from inside of the car. Therefore, to have good control strategy, the controller can use as many sensor feedbacks as possible. This calls for the observation enrichment. By observation enrichment, we mean to expand our observation capability. It does not necessarily call for the use of extra sensors.

For example, what we see in the rear view mirror about the car approaching from behind provides not only the distance information, but also the speed information (which is derived from the rate the distance is changing). This information can be used by the driver for speeding up the car or for changing lanes. This form of indirect feedback information (deriving both position and speed information from vision) does call for enhanced information processing capability. Some intelligent sensor capable of information processing should be a part of the intelligent controller. What has been discussed so far, such as controller input enrichment (observation enrichment of the plant), controller output enrichment (control enrichment of the plant), priority ranking in decision making (polarity over magnitude, qualitative over quantitative in the interpretation of error signal) are simpler to address than the more difficult problem of addressing the issue of correcting the error magnitude (the quantitative aspect). As we mentioned earlier, there seems to be a great deal of flexibility for reducing the error even after the error polarity has been determined. To reduce the error magnitude we have to work within the constraint of the control law. Too much and too little correction for the error magnitude is equally unsatisfactory. They may cause instability. Therefore, one basic consideration in controller design is to ensure the systems stability. Even if the control action results in a stable response, too much correction will cause overshoot and too little action will have a very sluggish response. So an intelligent controller should arbitrate stability and other performance conflicts.

To attain optimal performance in an ever changing environment, the controller ought to be time-varying and adaptive. Unfortunately, such time-varying controller can not be designed with conventional methods because system parameters model is lacking. It appears that a viable approach will be a heuristic, rule-driven controller that can be fine tuned.

III. Some Thoughts on Control Design Strategies

As was pointed out earlier, it is either impossible or impractical to have a precise dynamical model of a complex system. Therefore the control design strategies for such systems also calls for changes from conventional approaches which rely heavily on a purely deductive mathematical discipline. Given below are some of our thoughts on what should be when we design control systems.

1. Place less emphasis on the need of characterizing the system with a precise mathematical model with known parameters. Instead, attention should be focused on a better understanding of the system input/output (I/O) relationship. In other words, the system is to be characterized more explicitly in terms of its cause and effects relationship. More precisely speaking, it is more

important to have a good understanding of what the inputs will do to the outputs. So qualitative (i.e. physical) description of the system is more important than the quantitative (i.e. mathematical) description of the system. Mathematical model is needed mainly for the basic operation of the system, and hence, better to have the model as simple as possible. As such, lumped I/O characterization (e.g. qualitative cause-effect, relationship or transfer function) will be preferred to the state space characterization.

2. Single-input single-output (SISO) system design should be replaced by multiple-input, multiple-output (MIMO) design if needed through control enrichment and observation enrichment of the system. It is well known that an uncontrollable or an unobservable system can be made to be controllable and observable through expansion of control capability or observation capability. The multiple controls should be made to be as independent as possible. This control-decoupling property will render more transparent I/O characterization and hence facilitate the decision making of the controller. The multiple-observations of the system should be closely related to allow accurate and reliable assessment of the system state for intelligent decision making. In some applications, remote control and remote (non-contact) sensing (as in space, nuclear and robots applications) should also be explored. Indirect sensing (i.e. multiple information obtained from signal processing of a single observed information, e.g. vision may provide position as well as velocity information) may provide versatile and effective observation.

3. Future controllers can be an intelligent controller. The controller will have data base for storing commands (on task requirements) and rule-based algorithms (either preprogrammed or accumulated from experiences). It has memory for storing past information for learning from experiences. The decision making of the controller will be an expert system which consists of a collection of basic decision makers. The control law generated by the controller is a collective effort of each individual decision maker whose participation or not is decided intelligently in a time-varying manner based upon rule-based interpretations of the observed information and system requirements.

4. Rule-based decision making will be a combination of qualitative (logical) and quantitative (computational) nature. A mixture of heuristic as well as deductive decision processes. In a hierarchical decision process, the qualitative decision (i.e. which way the correction of error has to be) will have precedence over that of quantitative decision (i.e. how much the error correction is to be made). The quantitative decision will be a time-varying process. Instead of being a continuously varying process, a more practical way appears to be piecewise-constant time-varying process. In other words, the parameters of change are a set of finite values and their values will be changed at discrete instant of times decided by rule-based algorithms.

5. Rule-based algorithms are designed to ensure the stability of the system and are used to arbitrate conflicting performance requirements over different intervals of overall operation for maximizing the total performance.

6. An intelligent controller must be capable of learning from experiences and of optimizing performances in a given set of conditions. Therefore, an intelligent controller will be a combined effort of applying results from Artificial Intelligence (AI), Control and Signal Processing (CSP) and Operational Research (OR).

7. The advancements of computer technology are making intelligent controllers more feasible.

8. Integrated system design concept should be developed. To design an intelligent control system considerations for plant, control capability, (actuator technology) observation capability (sensor technology), and controller decision making all need to be addressed at the outset. Conventional control approach of starting with a given plant and then working to find a needed compensator (controller) to ensure satisfactory performance should be discouraged. Instead, we should always ask the

question of "how to have the 'best' control system ('best system' is not meant to be an optimal system in the traditional sense) for the tasks to be accomplished?"

9. We need better understanding on learning theory, time-varying and/or nonlinear control theory.

Shown in figures 1 and 2 are possible configurations of control systems with intelligent controllers.

IV. Examples

Given below are two examples on controller design based on combination of heuristic as well as deductive reasoning. They are found to produce much better performances than those designed with conventional design method.

Figure 3 shows the step response of a double-integrator that produces a fast response without causing any overshoot. The controller is a simple gain scheduling scheme designed by dynamic pole placement argument [10]. It is based on good qualitative understanding of the effects of velocity loop (that affects the damping) and the position loop (that affects the bandwidth) on the system response.

Figure 4 shows the improvement of convergence of a learning controller. The controller incorporates a combination of two separate learning algorithms operating over two different intervals of operations. The learning algorithm 1 applies only to linear system [11] which diverges when applied to a nonlinear system. The learning algorithm 2 [12] applies to some nonlinear system, but converges slowly. By using learning algorithm 1 for start up and then switching to algorithm 2, it is found that convergence rate for nonlinear system improves significantly.

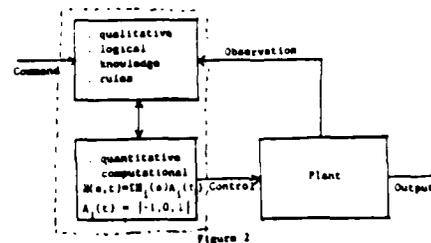
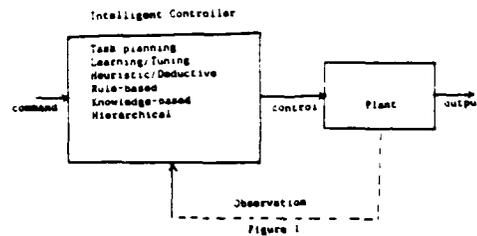
More and more intelligent controllers using expert systems have begun to emerge. Real-time expert systems for desalinization system and robot ping-pong game have been reported recently by Reliable Water Inc. and AT&T respectively. Without any doubt, intelligent control will be used more and more in the future for control of complex systems.

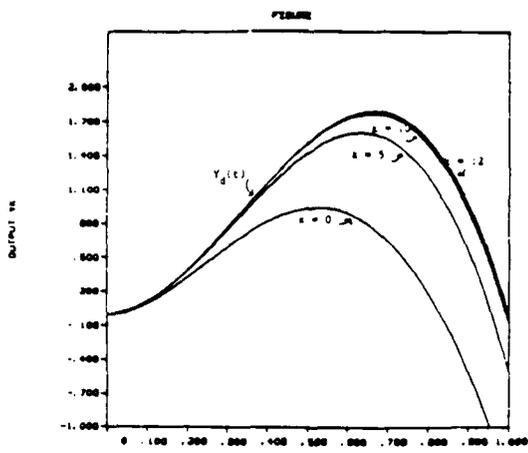
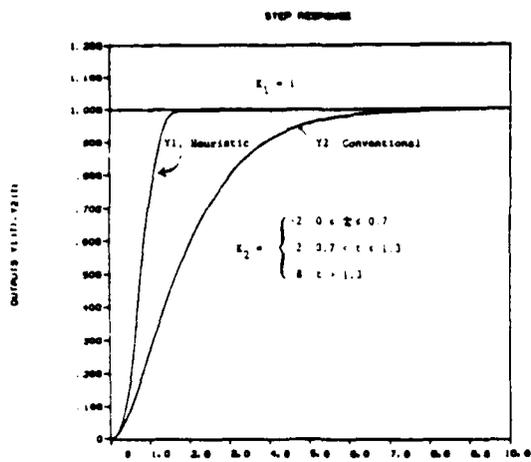
Acknowledgement

This work is supported in part by the Office of Naval Research under contract N00014-84-K-0425.

Reference

- [1] Kuo, B.C., "Automatic Control Systems" 5th ed., Prentice Hall, 1987
- [2] Horowitz, I.M., "Synthesis of Feedback Systems" Academic Press, 1963.
- [3] Truxal, J. G., "Control System Synthesis" McGraw Hill, 1955.
- [4] Kailath, "Linear Systems" Prentice Hall, 1980.
- [5] Astrom, K. J. and B. Wittenmark, "Computer Control Systems", Prentice-Hall, 1984.
- [6] Chen, C. T., "Linear System Theory and Design", Holt, Rinehart and Winston, 1984.
- [7] Kirk, D. E., "Optimal Control Theory", Prentice Hall, 1970.
- [8] Proceedings of IEEE Workshop on Intelligent Control 1985, August 26, 1985. RPI, Troy, New York.
- [9] Abstracts, 1987 IEEE International Symposium on Intelligent Control, January 18-20, 1987, Philadelphia, Pa.
- [10] Wu, M. Y., "On the Design of Time-varying Controllers for Producing Desired System Responses", Proc. of the 18th Asilomar Conference on Circuits, Systems and Computers, Nov. 4-7, 1984, Monterey, Ca.
- [11] Chouikha, M. F. and M. Y. Wu, "Learning Algorithm for Control Systems," to be published, the Proc. of 9th IASTED International Symposium, Robotics and Automation, May 27-29, 1987, Santa Barbara, Ca.
- [12] Arimoto, S., et.al., "Learning Control Theory for Dynamical Systems," Proc. 24th Control and Decision Conference, Ft. Lauderdale, Fla., Dec. 1985, pp. 1375-1380.





Learning algorithm for control systems*

M. F. Chouikha and M. Y. Wu
Department of Electrical and Computer Engineering
University of Colorado
Boulder, CO 80309-0425

Abstract

The case of controlling dynamical systems with unknown or partially known dynamics which are required to repeatedly track a desired trajectory over a finite time interval is of great interest in practice. One possible approach to solve such problem is the application of adaptation and learning in automatic systems. This paper presents a method of designing a learning controller based on the representation of the input-output signals by a set of orthogonal functions. Two theorems concerning the learning strategies in the design of the controller are given. They extend the results of S. Arimoto's betterment processes in the sense that some less restrictive conditions for convergence are needed. Examples are also given to illustrate the results. Potential applications to robot manipulator control are also noted.

1. Introduction

In practice, we often face the problem of controlling dynamical systems with incomplete or little a priori information about the plant. The solution is usually based on the application of adaptation and learning in automatic systems which reduced initial uncertainty by using the information obtained during the process of control. Due to the availability of an ever-increasing computational power, many adaptive and learning algorithms have emerged and numerous successful applications have been reported [1] and [10]. Recently a learning control algorithm called betterment process was proposed by S. Arimoto et al. [2] to solve the particular but in practice very important case of controlling systems with unknown or partially known dynamics which are required to repeatedly track a desired trajectory. The iterative approach taken by Arimoto et al. is based on a simple algorithm which updates the input to the system based on the previous input and a function of the difference between the previous output and the desired trajectory. However, this method is limited to a certain classes of systems and is applicable under certain rather restricted conditions. In this paper an alternative approach based on the representation of input-output signals by a set of orthogonal functions is presented. Orthogonal func-

tions such as the complex exponentials, the Walsh functions etc... are attractive primarily because of their potential of signal characterization with definite advantages in computational aspects [11].

2. Main results

In what follows we shall consider the linear time invariant single-input single-output system defined by: $y = L(u)$ where u, y are the system input and system output respectively and L is a linear operator. We also consider the following iteration formulae for input in the learning process:

$$u_{k+1}(t) = u_k(t) + \gamma e_k(t) \quad (1.1)$$

or

$$u_{k+1}(t) = u_k(t) + \delta \frac{d}{dt} \{e_k(t)\} \quad (1.2)$$

and the error

$$e_k(t) = y_d(t) - y_k(t) \quad (1.3)$$

where $u_k(t)$ and $y_k(t)$ are respectively the input and the output of the system for the k -th iteration, $y_d(t)$ is some given function and γ, δ are some constants. The problem is then to find conditions that ensures

$$y_k(t) \rightarrow y_d(t) \text{ as } k \rightarrow \infty$$

in some sense, for all t i.e.

$$\|e_k(t)\| \rightarrow 0 \text{ as } k \rightarrow \infty \text{ for all } t$$

for some chosen norm.

Definition

The function $\tilde{f}(t) = \tilde{f}(t, N, \Phi)$ defined in the interval $[t_1, t_2]$ is said to be an ϵ -approximation of a given function $f(t)$ on the interval $[t_1, t_2]$, with respect to a chosen set of orthogonal functions described by the vector valued function:

$$\Phi = [\phi_1(t), \phi_2(t), \phi_3(t), \dots, \phi_n(t)]^T$$

if for a given positive ϵ there exists a number N such that for all t in $[t_1, t_2]$

$$\|f - \tilde{f}\| \leq \epsilon \text{ for } n \geq N$$

In this paper $\|g\|$ denotes the L2-norm of a square-integrable function $g(t)$ defined on the interval $[t_1, t_2]$ by

$$\|g\| = \left(\int_{t_1}^{t_2} g^2(t) dt \right)^{1/2} \quad (1.4)$$

* This work is supported in part by the Office of Naval Research under contract N00014-84-K-0425

Lemma

Let the real valued function $f(t)$ be square-integrable on the interval $[t_1, t_2]$ then the ϵ -approximation of $f(t)$ is given by

$$\hat{f}(t) = \sum_{i=1}^N c_i \phi_i(t) \quad (1.5)$$

where

$$c_i = \gamma_i / k_i \quad i = 1, 2, \dots, N \quad (1.6)$$

$$\gamma_i = \int_{t_1}^{t_2} f(t) \phi_i(t) dt \quad (1.7)$$

$$k_i = \int_{t_1}^{t_2} \phi_i^2(t) dt$$

The main results of the learning process will be given in the following two theorems without proof. The proof can be found in [9].

Theorem 1.

Let $y_d(t)$ be a given desired trajectory defined over a time interval $[t_1, t_2]$ and let $e_k(t)$ be as defined in (1.3) and $\hat{e}_k(t)$ be its ϵ -approximation with respect to Φ . Also let

$$L(\Phi) = P\Phi \quad (1.8)$$

where P is a $N \times N$ constant matrix, and L is the linear operator representing the system. Then

$$\|\hat{e}_k\| \rightarrow 0 \text{ as } k \rightarrow \infty \quad (1.9)$$

if and only if

$$(\text{tr}(P))^2 > (N-1)\|P\|_F^2 \quad (1.10)$$

where $\|P\|_F^2$ and $\text{tr}P$ are the Frobenius norm and the trace of P respectively.

It is noted that condition (1.10) is somewhat restrictive and may not be satisfied for certain class of linear systems. This condition can be relaxed by the following theorem.

Theorem 2.

For the problem considered in theorem 1 and a given desired trajectory $y_d(t)$ defined over the time interval $[t_1, t_2]$, and for a given set of orthogonal functions defined by the vector valued function Φ if there exists a constant $N \times N$ matrix Q which satisfies the following condition

$$\|I - QP^T\|_F < 1 \quad (1.11)$$

then

$$\|\hat{e}_k\| \rightarrow 0 \text{ as } k \rightarrow \infty$$

Comments

1. The generalization of theorem 1 and theorem 2 to the case of multi-dimensional linear systems is straight forward.
2. The computational complexity of the learning algorithm may depend on the types of the orthogonal functions chosen in the ϵ -approximation.
3. If P is nonsingular then the best choice for Q is

$$Q = P^{-T}$$

4. It has been shown [2] that using the feedback linearization technique an n -joint robot manipulator can be transformed to a set of n decoupled second order linear time-invariant systems. therefore the learning algorithm presented in this paper can also be applied to a certain class of nonlinear systems, such as robot manipulator control systems.

3. Examples

The illustration of the two theorems presented in this paper is given in the following two examples :

Example 1

The applicability of the result of theorem 1 is illustrated in this example. Consider the first order linear time invariant system described by the following transfer function:

$$F(s) = 100 \frac{s + .89}{s + 0.9}$$

For easy checking with analytic result let the desired output $y_d(t)$ be

$$y_d(t) = 3.46410t - 0.73205$$

which can also be written as in (1.5)

$$y_d(t) = c_1 \phi_1(t) + c_2 \phi_2(t)$$

where

$$c_1 = 1$$

$$c_2 = 1$$

and

$$\phi_1(t) = 1$$

$$\phi_2(t) = 2\sqrt{3}t - \sqrt{3}$$

are two Legendre functions. In this case $N = 2$ and

$$P = \begin{bmatrix} 99.62 & -19 \\ .19 & 99.94 \end{bmatrix}$$

the condition (1.10) is satisfied and the algorithm (1.1) converges in 2 iterations as it is shown in figure 1.

Example 2

Consider the following system:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (2.1)$$

where

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [1 \quad 0]$$

and the desired output is

$$y_d(t) = 12t^2(1-t) \quad (2.2)$$

for $t \in [0, 1]$. For this system the condition (1.10) of theorem 1 is not satisfied, however by choosing a set of 4 Legendre functions with a matrix Q

$$Q = \begin{bmatrix} -100.58 & 177.64 & -166.63 & 190.80 \\ -176.70 & 280.0 & -272.10 & 302.15 \\ -165.80 & 270.66 & -304.80 & 311.0 \\ -190.70 & 299.10 & -308.60 & 230.50 \end{bmatrix}$$

for which condition (1.11) is satisfied, $\hat{y}_k(t)$ converges to $y_d(t)$ in 4 iterations as it is shown in figure 2. We notice also that since in this example $CB = 0$, the conditions for the convergence of the C^1 -type betterment as has been shown in [3], [4] are not satisfied. Figure 3 shows the divergence of $y_k(t)$ from the desired trajectory $y_d(t)$ as the number of iteration k gets larger. However, for a system with the same matrices A and B as the system considered in this example but with matrix C equal to :

$$C = [0 \quad 1]$$

the C^1 -type betterment converges as it is shown in figure 4.

References

- 1] Tsytkin, Y.Z. "Adaptation and learning in automatic systems" Academic Press, New York 1971.
- 2] Arimoto, S. *et al.* "Learning control theory for dynamical systems" 24th Conference on Decision and Control FLORIDA December 1985 pp 1375-1380.
- 3] Arimoto, S. *et al.* "Application of learning method for dynamic control of robot manipulators." 24th Conference on Decision and Control FLORIDA December 1985 pp 1381-1386.
- 4] Fu, K.S. "Learning control systems -Review and outlook" IEEE transactions on automatic control, April 1970 pp 210-221.
- 5] Ljung, L., and Soderstrom, T. "Theory and practice of recursive identification" The MIT Press, Cambridge, Massachusetts 1983.
- 6] Mendel, J. "Discrete techniques of parameter estimation - the equation error formulation" Marcel Dekker, Inc. New York 1973.
- 7] Arimoto, S. *et al.* "Betterment operation of dynamical systems by learning a new control theory of servomechanisms or mechatronics" Proc. 23 rd IEEE LAS VEGAS Dec 1984.
- 8] Kailath, T. "Linear systems" Prentice - Hall, Inc. New Jersey 1980.
- 9] Choukha, M.F. "Learning in automatic control" Phd thesis to be published Dec 1987.
- 10] Mita, T., and Kato, E. "Iteration control of robot manipulator" Proc. 15th ISIR Tokyo Japan 1985.
- 11] Rao, G.P. "Piecewise constant orthogonal functions and their application to systems and control" Springer-Verlag 1983.
- 12] Wu, M.Y., and Arbouz, N.M. "Controller design of a multi-joint robot Manipulator" Proc of the 9th IASTED International symposium on Robotics and automation May 1985 pp 54-58

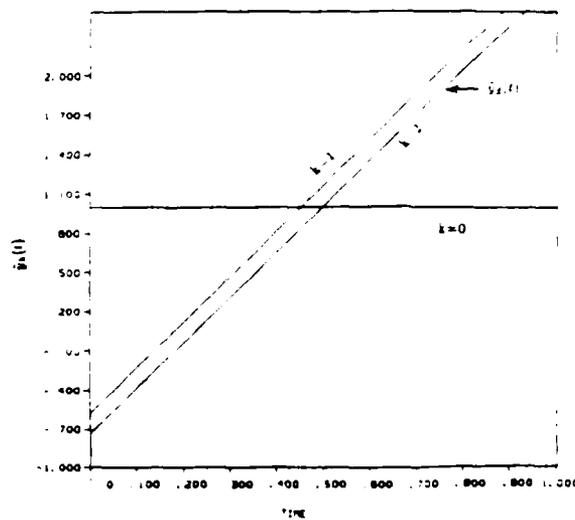


Figure 1

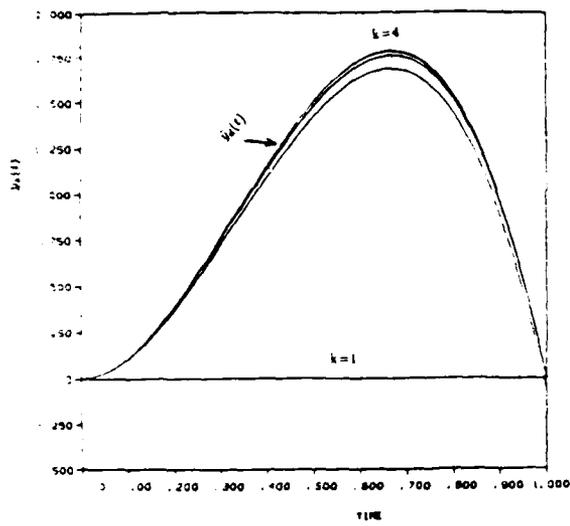


Figure 2

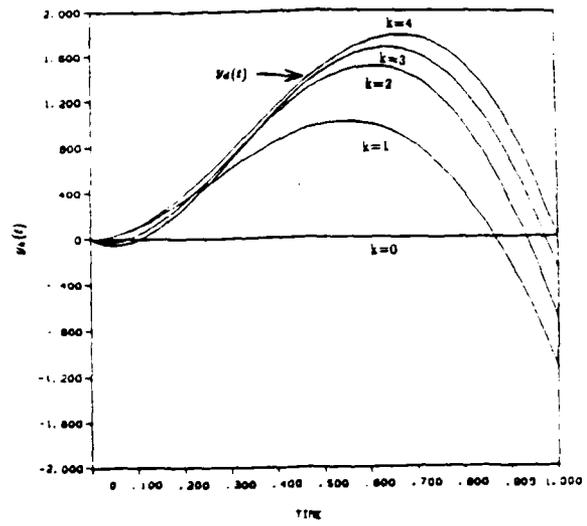


Figure 4

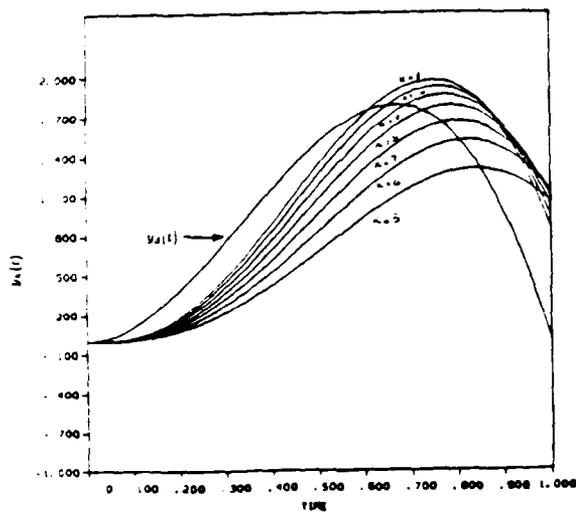


Figure 3

END

FILMED

FEB. 1988

DTIC